Fachhochschule Köln
Abteilung Gummersbach

Fachbereich Informatik
Studiengang Wirtschaftsinformatik

Diplomarbeit

Konzeption und Entwicklung eines Oracle-basierten Innovationscontrolling in einer Forschungseinrichtung

(Stiftung caesar, Bonn)

Eingereicht von

Johannes Schmidt
Matrikelnummer: 102 345 51

Prüfer:

Frau Prof. Dr. Heide Faeskorn - Woyke
Herr Thomas Wagner, Stiftung caesar

Bonn, 10. Mai 2001

Danksagung

Vor allem danke ich meiner betreuenden Professorin Frau Faeskorn-Woyke und meinem zweiten Betreuer Herrn Thomas Wagner. Ausserdem bedanke ich mich herzlich bei meiner Familie, die mich während des Studium sehr unterstützt hat. Mein besonderer Dank gilt meiner Frau Helene, die mir während meines Studiums beigestanden hat.

Ehrenwörtliche Erklärung:

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Bonn, 10. Mai 2001 Johannes Schmidt

Teil A: Theoretischer Teil

1. Einleitung

1.1 Einführung in die Thematik der Diplomarbeit

Ziel dieser Diplomarbeit ist es, eine Datenbankanwendung für eine öffentliche Stiftung aufzubauen, in der Projektideen erfasst und bewertet werden können.
Da bei der Stiftung caesar die Anwendung *Oracle Applications* eingesetzt wird, werden für die Realisierung der Datenbankanwendung Oracle-Werkzeuge verwendet. Diese werden im Kapitel 5: Entwicklungswerkzeuge und Kapitel 6: *Oracle Applications* beschrieben.

1.2 Vorgehen und Gliederung

Die Diplomarbeit gliedert sich in folgenden Kapitel:

Teil A: Theoretischer Teil

- Kapitel 1: Einleitung
- Kapitel 2: Beschreibung der Stiftung caesar
- Kapitel 3: Innovationscontrolling
- Kapitel 4: Datenbankentwicklung sowie das Pflichtenheft zu InnoCon
- Kapitel 5: Entwicklungswerkzeuge
- Kapitel 6: Einführung in *Oracle Applications*, Überblick über *Oracle Financials*

Teil B: Praktischer Teil

- Kapitel 7: Aufbau einer eigenen Anwendung in Oracle Applications
- Kapitel 8: InnoCon

Teil C: Anhang

- Kapitel 9: Quellcode, Literaturquellen

Im theoretischen Teil wird das Innovationscontrolling behandelt. Danach werden die Datenbankentwicklung, die benutzten Entwicklungswerkzeuge und *Oracle Applications*, in denen die Anwendung InnoCon implementiert wurde, erläutert.
Der Name **InnoCon** wurde vom Autor ausgewählt, damit der Leser sowie die Benutzer der entwickelten Anwendung die Thematik, die beschrieben wird, sofort erfassen. Der Name InnoCon leitet sich aus dem Begriff „Innovationscontrolling“ ab.

Im praktischen Teil wird die Realisierung der Anwendung InnoCon beschrieben.
Im Anschluss wird auf die weitere Verwendung von InnoCon eingegangen. Im Anhang sind die Literaturquellen und der Quellcode aufgelistet.

2. Stiftung caesar

Gründung von caesar

caesar ist als allgemeine selbständige Stiftung im Rahmen der Ausgleichsmaßnahme und des Bonn-Berlin-Gesetzes vom 26. April 1994 zur Unterstützung des Strukturwandels gegründet worden. Der Name caesar ist eine Abkürzung für center of advanced european studies and research.

Die Forschungsbereiche von caesar umfassen die Gebiete Smart Materials und Mikromechanik.

Für die Durchführung der Forschungsprojekte steht den Forschungsgruppen ein bestimmtes Budget zur Verfügung, für das die jeweilige Gruppe bzw. deren Leiter verantwortlich ist und mit dem sämtliche für das Projekt anfallende Aufwände bestritten werden müssen. Die Forschungsarbeit wird in Forschungs- und Projektteams aufgeteilt, die Forschungsprojekte sind voraussichtlich auf eine Laufzeit von maximal fünf Jahren beschränkt.

Die Stiftung arbeitet in verschiedenen Kooperationsformen mit Universitäten und Wirtschaftsunternehmen zusammen. In diesem Rahmen handelt die Stiftung selbst als Unternehmen, nimmt Aufträge entgegen, erbringt Leistungen und stellt entsprechende Rechnungen.

Research in Triplets

Bei caesar arbeiten jeweils drei Forschungsteams an einem Forschungsprojekt gemeinsam. Dieses Dreierteam wird als ein sogenanntes „Triplet“ bezeichnet.

Durch die drei Arbeitsgruppen, die im folgenden beschrieben werden, wird jeder thematische Schwerpunkt aus unterschiedlichen Blickwinkeln bearbeitet.

Hauptsächlich Informatiker, Mathematiker und grundlagenorientierte Naturwissenschaftler sorgen für die Verankerung der Forschungsarbeit durch Modellbildung und unterstützen durch Simulationen die Versuchsplanung. Dieses Aufgabengebiet betrifft die sogenannte *Modellgruppe*.

Alle notwendigen Versuche und die dazugehörigen Auswertungen führen in der Regel die Naturwissenschaftler durch. Man bezeichnet diese Gruppe als *Experimentalgruppe* .
Für die konkrete Umsetzung der Ergebnisse in ein marktfähiges Produkt sorgt ein Team von anwendungsorientierten Ingenieuren und Naturwissenschaftlern in einer *Ingenieurgruppe.*

Um ein Gelingen der Projekte zu gewährleisten, ist die Kooperation der Gruppen unverzichtbar. Leiter jeder Arbeitsgruppe ist ein Wissenschaftler. Die drei Arbeitsgruppenleiter eines Themenschwerpunktes (Triplets) übernehmen deren Gesamtleitung gemeinschaftlich und gleichberechtigt.

Themenfindung

In der Aufbauphase werden die Schwerpunkte bei caesar vorwiegend durch die Arbeitsgebiete des wissenschaftlichen Vorstandes bestimmt. Unternehmen sowie universitäre und außeruniversitäre Forschungsinstitutionen und die Partner des regionalen Netzwerkes sind ein anderer wichtiger Impulsgeber.
Ein wichtiges Ziel der Stiftung caesar ist es, die Arbeitsgruppen durch Fachleute aus Industrie und Wirtschaft (industrial fellows) zu verstärken. Wie oben erwähnt sind auch diese Formen der Zusammenarbeit allein am Erfolg des Projektes orientiert. [1]

[1] Vgl. Hoffmann: "Mission von caesar" Stiftung caesar, Bonn 2000

Wissenschaftliche Projekte bei caesar

Wie eingangs bereits erläutert, wurde die Stiftung caesar mit dem Ziel gegründet, Forschungsprojekte zu realisieren.

caesar betreibt weder ausschließlich Grundlagenforschung noch reine Auftragsforschung, im Mittelpunkt der Projektauswahl steht die Orientierung am Markt. Drei Schwerpunktbereiche bestimmen die Ausrichtung der Projekte. Diese sind:

1. Materialwissenschaften / Nanotechnologie,
2. Koppelung biologischer und elektronischer Systeme sowie
3. Kommunikationsergonomie.

Im folgenden werden einige Forschungsprojekte aus den obengenannten Bereichen vorgestellt.

1. Materialwissenschaften/Nanotechnologie

Zu dem Bereich „Materialwissenschaften / Nanotechnologie“ gehört das Triplet „Smart Materials“ (Multifunktionale Schichten). Die drei Arbeitsgruppen dieses Triplets forschen und entwickeln Werkstoffe, die als „dünne Schichten“ bezeichnet werden. Diese Werkstoffe zeichnen sich dadurch aus, dass sie ihre Gestalt durch äußere Einflüsse, wie magnetische-, elektrische- oder Wärme-Energie, verändern können.

Eine Art der Smart Materials ist der Werkstoff mit Formgedächnis. Formgedächtniswerkstoffe sind Metall-Legierungen, die nach dem Verbiegen wieder ihre ursprüngliche Form annehmen, sobald man sie auf eine bestimmte Temperatur erwärmt. caesar entwickelt Formgedächtnis-Materialien, die z.B. für Zahnspangen verwendet werden können, die nur noch selten nachjustiert werden müssen.

2. Koppelung biologischer und elektronischer Systeme

Die Projektgruppe „Implant Materials“ gehört zu dem Bereich „Kopplung biologischer und elektronischer Systeme“. In dieser Projektgruppe werden bioverträgliche Implantatmaterialien entwickelt, die bakterielle Infektionen und Abstoßungsreaktionen bei den Patienten minimieren sollen.

3. Kommunikationsergonomie

Das Triplet „Computergestützte Chirurgie“ beschäftigt sich mit der Entwicklung und Herstellung von Software- und Hardwaresystemen für die bildgeführte Chirurgie sowie deren Anwendung in der Mund-, Kiefer- und Gesichtschirurgie, Neurochirurgie und Orthopädie. Der Forschungsbereich liegt hier in der Virtual-Reality-Techniken, um computergestützte Operationen mit Hilfe von Laser-Technik zu ermöglichen.

Dieses Triplet besteht aus den folgenden drei Arbeitsgruppen:

Die erste Arbeitsgruppe „Surgical Simulation and Navigation“ (Modellgruppe) entwickelt Softwaresysteme zur Chirurgiesimulation.

In der Arbeitsgruppe „Holografie und Lasertechnologie“, der Experimentalgruppe, werden holografische Verfahren mit Lasertechnologien entwickelt. Diese dienen der exakten Gesichtsprofilvermessung von Patienten, die sich einer Kiefer- oder Gesichtsoperation unterziehen müssen.

In der dritten Arbeitsgruppe “Rapid Prototyping and Surgical Instruments“, der sogenannten Ingenieurgruppe, werden anatomische Modelle und Implantate entwickelt.[2]

Der Kern aller Aktivitäten bei caesar sind die wissenschaftlich-innovativen Projekte. Im folgenden Kapitel wird erläutert, was Innovationscontrolling ist und welche Werkzeuge und Analysetechniken für das Innovationscontrolling verwendet werden.

[2] Vgl. Hugenroth: „Forschen in neuen Dimensionen“, Stiftung caesar, Bonn 2000

3. Innovationscontrolling

Um sich mit dem Controlling eines Aufgabengebietes beschäftigen zu können, ist es notwendig, zunächst das entsprechende Aufgabengebiet zu definieren. In diesem Abschnitt wird eine Trennung der Begriffe Innovation und Controlling zum besseren Verständnis vorgenommen.

3.1 Definition und Abgrenzung des Innovationsbegriffs

Der Begriff Innovation stammt von dem lateinischen Wort „innovo“ ab, was soviel wie Neuerung, Erneuerung, Neueinführung oder auch Neuheit bedeutet.[3] Als grundlegendes Kriterium für eine Innovation gilt demnach die Neuartigkeit, wobei allerdings zwischen Innovation und Invention unterschieden werden muss.

Invention

Die **Invention**, oder auch Erfindung, ist die notwendige Vorstufe der Innovation. Sie beschränkt sich auf den Prozess der Wissensgenerierung durch Forschung und Entwicklung und die erstmalige technische Realisierung. Sie kann geplant aber auch zufällig sein.[4]

Innovation

Werden neuartige Produkte oder Verfahren geschaffen und auf dem Markt oder im Betrieb eingeführt, so spricht man von **Innovation**.[5] Der Begriff der Neuartigkeit bezieht sich dabei allein auf das betrachtete Unternehmen. Eine Produktentwicklung kann somit für eine Unternehmung eine Innovation darstellen, obwohl dieses Produkt bereits in einem oder mehreren Konkurrenzunternehmen entwickelt wurde.

[3] Vgl.: Stippel: Innovationscontrolling, Seite 6.
[4] Vgl.: Stippel: Innovationscontrolling, Seite 6.
[5] Vgl.: Stippel: Innovationscontrolling, Seite 7.

Im Gegensatz zur Invention ist die Innovation als die erstmalige wirtschaftliche Anwendung einer neuen Problemlösung zu verstehen, das heißt, hier geht es um die ökonomische Optimierung der Wissensverwertung. Im Unterschied zu der zeitpunktbezogenen Invention ist die Innovation das Ergebnis eines Prozesses oder sogar der Prozess selbst, der alle Phasen von der Ideengewinnung über die Ideenumsetzung bis hin zur Ideenverwertung enthält. [6]

3.1.1 Phasen der Innovation

Die Zeitspanne, die ein Innovationsprozess umfasst, lässt sich in verschiedene Phasen unterteilen. In Abbildung 1: werden alle Aktivitäten, die nötig sind, um von einer Idee zu ihrer praktischen Umsetzung gelangen zu können, dargestellt:

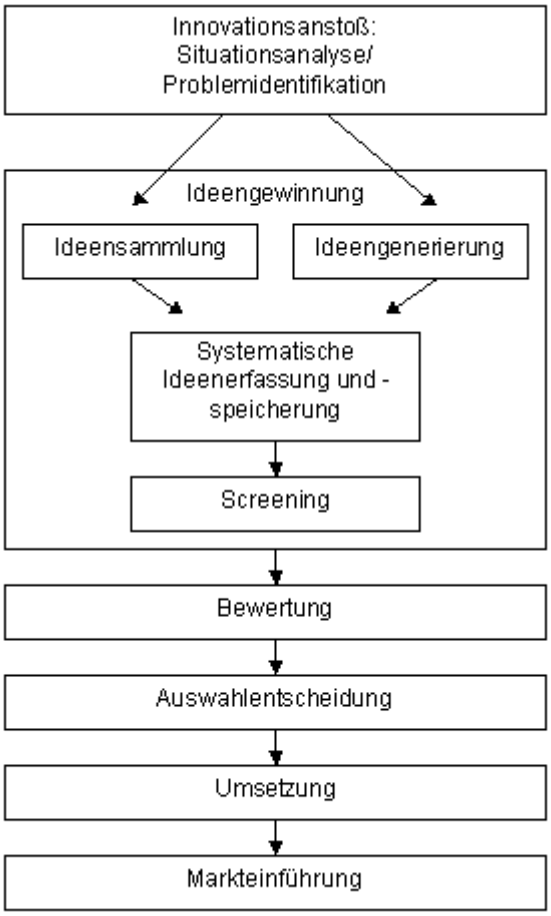


Abbildung 1: Grundschema eines Innovationsprozesses

[6] Vgl. Stippel: „Innovationscontrolling“ Seite 6 f.

3.1.1.1 Ideengewinnung

Nicht alle gesammelten oder neu gewonnenen Ideen und Vorschläge stellen Lösungsmöglichkeiten für das konkrete Problem dar. Oft entstehen im Rahmen der Ideensammlung und -gewinnung auch Vorschläge für andere Problembereiche. Diese sollten nicht ohne weiteres eliminiert werden. Viel eher ist es sinnvoll, die Ideen in einer Ideendatenbank zu speichern, um auf diese dann möglicherweise zu einem späteren Zeitpunkt im Rahmen der Ideensammlung zurückgreifen zu können. Um die ursprüngliche Problemstellung nicht aus den Augen zu verlieren, schließt sich an die Ideenerfassung ein sogenanntes *Screening* an. Das Screening gewährleistet, dass in der Bewertungsphase nur solche Ideen einer eingehenden Prüfung und Beurteilung unterzogen werden, die grundsätzliche Erfolgschancen aufweisen. Dadurch wird die in den vorherigen Phasen durchaus gewollte Ideenflut unter Zweckmäßigkeitsaspekten verringert.[7]

3.1.1.2 Bewertung von Ideen

Die Bewertung einer Projektidee bedeutet, dass diese hinsichtlich ihrer technischen Umsetzbarkeit, ihres voraussichtlichen Markterfolgs und ihres Beitrags zu den angestrebten Zielsetzungen beurteilt wird. Um subjektive Fehleinschätzungen und die damit verbundenen finanziellen Risiken für das Unternehmen zu vermeiden, sind zuverlässige Entscheidungsgrundlagen und -methoden von großer Bedeutung. [8]

[7] Vgl. Boutellier, Völker, Voit: „Innovationscontrolling“ Seite 159.
[8] Vgl. Boutellier, Völker, Voit: „Innovationscontrolling“ Seite 160.

3.1.1.3 Ideenauswahl

Unabhängig von der Ideenbewertung ist die Auswahl der umzusetzenden Alternativen zu betrachten. Die Auswahlentscheidung kann durchaus die Aufgabe einer anderen Instanz sein als derjenigen, die mit der Bewertung der Vorschläge beauftragt wurde. Die eigentliche Entscheidung, ob eine neuartige Idee auch tatsächlich realisiert wird, obliegt dann der Unternehmensführung, also dem Top-Management, das schließlich für den wirtschaftlichen Erfolg oder Misserfolg verantwortlich ist.[9]

3.1.1.4 Ideenumsetzung / Ideenrealisierung

Nach der Entscheidung wird die ausgewählte Alternative in die Praxis umgesetzt. In der Phase der Ideenumsetzung sollten die zuvor ausgewählten Produktideen zu marktfähigen und wirtschaftlich erfolgreichen Produkten entwickelt werden. Dazu bedarf es einer präzisen sach-, termin- und kostengerechten Planung und Realisierung des Innovationsvorhabens.
Ideen, die zwar als solche neu sind, deren Art, Bezugsbereich und Umfeld allerdings schon im Unternehmen bekannt sind, lassen sich häufig in Routineprozessen mit relativ geringen Anpassungen an die Besonderheiten der jeweiligen Idee realisieren. In solchen Fällen spricht man von **Innovationsroutine**.

3.1.1.5 Markteinführung

In der Phase der Ideenrealisierung werden die ausgewählten Neuerungen umgesetzt. Den Abschluss bildet die Markteinführung und die erfolgreiche Vermarktung des Produktes. Hierbei handelt es sich also neben der technischen auch um die kaufmännische Umsetzung des geplanten Konzeptes.

[9] Vgl. Boutellier, Völker, Voit: „Innovationscontrolling“ Seite 160.

Die eigentliche Innovation findet erst in dieser Phase statt, da erst mit der konkreten Umsetzung einer Invention zu einem marktfähigen Produkt etwas Neues entstanden ist, das auf einem Markt angeboten wird. Ein besonders wichtiger Aspekt ist hierbei die Sicherung des geistigen Eigentums einer Invention, sofern sie eine objektive Innovation darstellt.[10] Das Patentrecht ist an dieser Stelle von großer Bedeutung, da sonst eine Unternehmung Gefahr läuft, ihre Innovation der Konkurrenz zur Verfügung zu stellen und dadurch den angestrebten und auch erarbeiteten Wettbewerbsvorteil wieder zu verschenken.

3.1.2 Innovationsarten

Generell unterscheidet man drei Arten von Innovation:

- Prozess- oder auch Verfahrensinnovation
- Produktinnovation
- Sozialinnovation

3.1.2.1 Prozess- / Verfahrensinnovation

Prozess- oder auch Verfahrensinnovationen sind geplante Änderungen im Arbeitsablauf, die durch den bestehenden Wettbewerb und den damit verbundenen technischen Fortschritt erforderlich sind. Sie beeinflussen vor allem Qualität, Durchlaufzeiten und Kostenaspekte eines oder mehrerer Produkte.
Durch Verfahrensinnovationen können Wettbewerbsvorteile durch Einsparungen von Kosten bei der Produktion und durch den damit sinkenden Produktpreis und die daraus resultierenden Wettbewerbsvorteile durch kürzere und flexiblere Reaktionsmöglichkeiten und höhere Produktzahlen erreicht werden.[11]

[10] Vgl. Stippel: „Innovationscontrolling“ Seite 25.
[11] Vgl. Stippel: „Innovationscontrolling“ Seite 9.

3.1.2.2 Produktinnovation

Von Produktinnovation spricht man, wenn ein Unternehmen ein Produkt auf den Markt bringt, das bisher nicht im Produktionsprogramm dieses Unternehmens enthalten war. **Produkt** wird in dieser Definition gleichbedeutend zum Begriff des Wirtschaftsgutes verstanden. Hierdurch werden auch immaterielle Güter in die Definition eingeschlossen.
Ein Produkt kann z.B. hinsichtlich Qualität, Service, Funktionalität und Ergonomie verbessert oder neu gestaltet werden, so dass es unter konkurrierenden Produkten bessere Absatzchancen bekommt. Durch eine Produktinnovation kann sowohl eine Erweiterung des bestehenden Produktprogramms erreicht als auch neue Absatzmärkte erschlossen werden.[12]

3.1.2.3 Sozialinnovation

Sozialinnovationen beziehen sich auf die Steigerung der Leistungsfähigkeit und die Leistungsbereitschaft der Mitarbeiter in allen betrieblichen Aufgabenbereichen. Unter Sozialinnovation versteht man alle Neuerungen in einer Organisation, die nicht unter die technischen Innovationen fallen. Sie sind Neuerungen, die die Organisation als strukturelles und soziales Gefüge betreffen.[13]

[12] Vgl. Stippel: „Innovationscontrolling“ Seite 9.
[13] Vgl. Stippel: „Innovationscontrolling“ Seite 9.

3.2 Controlling

3.2.1 Definition des Begriffs Controlling

In die deutsche Sprache wurde der Begriff **Controlling** als Ableitung des englischen „to control“ eingeführt. *Control* in diesem Sinne bedeutet ein Steuern und Lenken des Unternehmens.

Das Controlling beinhaltet demnach die Planung und Steuerung der Betriebsprozesse und dient als Informations- und Führungsinstrument für das Unternehmen. Es soll dem Management die für die Wahrnehmung der Führungsfunktion relevanten Informationen bereitstellen, aufbereiten und interpretieren. Entscheidungen werden den Führungskräften durch das Controlling nicht abgenommen, sie sollen jedoch erleichtert werden.

Die speziellen Aufgaben und der Umfang des Controllings sind je nach Art des Wirtschaftszweiges und der Betriebsgröße unterschiedlich. Im Bank- oder Handelsbetrieb tauchen wesentlich andere Kontrollprobleme auf als in einer Forschungseinrichtung.

Der Betriebsprozess einer Forschungseinrichtung ist die Durchführung von Forschungsprojekten. Deswegen liegt hier ein besonderer Schwerpunkt im Forschungs- und Entwicklungs-Controlling (F+E-Controlling).

Der F&E-Prozeß stellt die erste Phase der Wertschöpfungskette dar, in der die Unternehmensstrategie in Form von konkreten Projekten physisch greifbar und im Sinne des Controllings messbar wird.[14]

[14] Vgl. Wöhe: „Einführung in die allgemeine Betriebswirtschaftslehre BWL“, Seite 175.

3.2.2 Controllingarten

Controlling als Konzept zur Steuerung der Wertschöpfung im Unternehmen untergliedert sich in operatives und strategisches Controlling.
An den beiden Steuerungsfragen: “Machen wir das Richtige und machen wir es richtig?“ erkennt man den Unterschied zwischen dem strategischen und dem operativen Controlling. Die Frage: “Machen wir das Richtige?“ ist die strategische, und die Frage: “Machen wir es richtig?“ die operative Komponente.

3.2.2.1 Strategisches Controlling

Das strategische Controlling ist auf einen längeren Zeitraum ausgerichtet und fokussiert längerfristige und unternehmensglobale Entwicklungen. Hier sind sämtliche Größen relevant, die für die Sicherung der dauerhaften Existenz des Unternehmens von Bedeutung sind.
Das strategische Controlling arbeitet nicht mit quantitativen Daten aus dem Rechnungswesen, sondern mit qualitativen Daten, beispielsweise mit Marktanteilsdaten oder der Verschiebung von Imagepositionen des Unternehmens.[15]

3.2.2.2 Operatives Controlling

Als Ergänzung des strategischen Controllings gilt das operative Controlling. Die Ausrichtung erfolgt auf interne Aspekte der Unternehmung.
Das operative Controlling grenzt sich vom strategischen Controlling durch seine Orientierung (Wirtschaftlichkeit innerbetrieblicher Prozesse), der Planungsstufe (taktische und operative Budgetierung), der Dimension (Aufwand - Ertrag und Kosten - Leistungen) und seinen Zielgrößen (Gewinn / Rentabilität) ab. Operatives Controlling hat zum Ziel, eine kurz- und mittelfristige Gewinnorientierung zu erreichen.[16]

[15] Vgl. „Lexikon des Controlling“ , München 1997.
[16] Vgl. Horvath: „Controlling”, Seite 145.

Horváth unterscheidet zwischen strategischem und operativem Controlling nach folgenden Merkmalen:

	Strategisches Controlling	**Operatives Controlling**
Orientierung	Umwelt und Unternehmung: Adaption	Unternehmung: Wirtschaftlichkeit betrieblicher Prozesse
Planungsstufe	Strategische Planung	Taktische und operative Planung, Budgetierung
Dimension	Chancen/Risiken, Stärken/Schwächen	Aufwand/Ertrag, Kosten/Leistungen
Zielgrößen	Existenzsicherung, Erfolgspotential	Wirtschaftlichkeit, Gewinn, Rentabilität

Tabelle 1: strategisches und operatives Controlling[17]

Bereichscontrolling

Eine Unterteilung der Aufgaben und Instrumentarien des Bereichscontrollings Forschung und Entwicklung kann nach Horváth wie folgt aussehen:

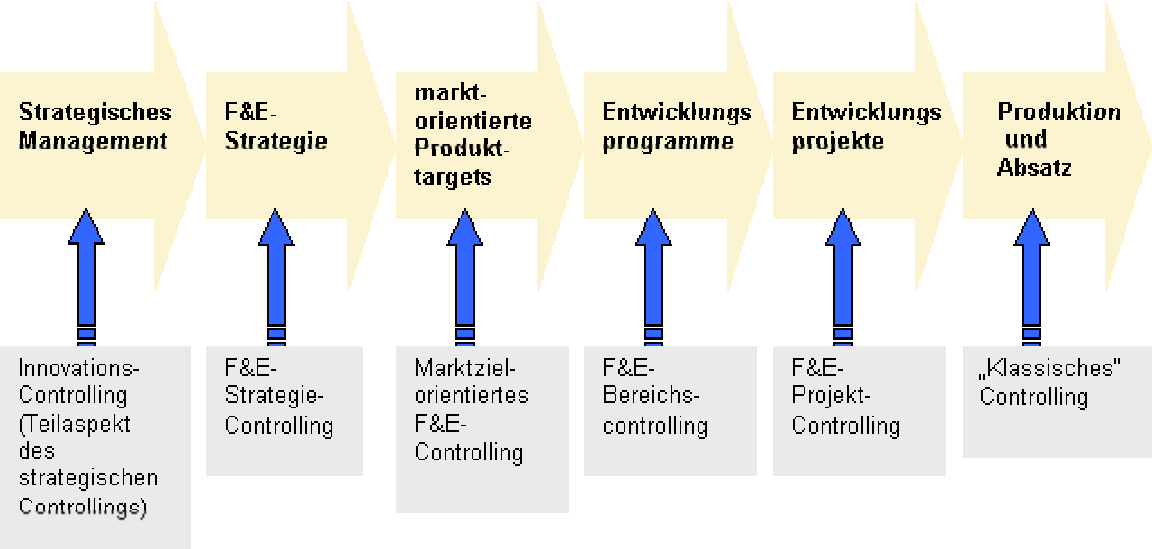


Abbildung 2: Ausbaustufen des F&E-Controllings[18]

[17] Vgl. Horváth: „Controlling“, Seite 250 Abb. 3.5.1.
[18] Vgl. Horváth: „Controlling“, Seite 840 Abb. 7.2.7.

Des öfteren, wenn von F&E-Controlling gesprochen wird, wird es als das eigentliche Innovationscontrolling verstanden. Deshalb ist eine Trennung der Begriffe oft schwierig: "Das operative Innovationscontrolling kümmert sich um die Abwicklung von F+E-Projekten. Die Frage, ob die heutigen Innovationsprojekte genügen, das Unternehmen in der Zukunft zu tragen, versucht das strategische Controlling zu beantworten."[19]

3.2.3 Innovationscontrolling

Innovationen sind von großer Bedeutung. Es gilt, stetig die Kosten zu senken und gleichzeitig die Marktleistung zu erhöhen. Als Grundlage für die Hervorbringung von Innovationen gelten die F&E–Bereiche, welche als das Hauptziel der Unternehmung eine langfristige Sicherung der Überlebens- und Entwicklungsfähigkeit gewährleisten sollen. Der Weg von der Innovationsidee bis zu einer erfolgversprechenden Markteinführung beinhaltet eine Fülle hochkomplexer Managementaufgaben und vielfältige Konfliktpotentiale. Ziel des Innovationscontrolling ist es, diesen Prozess bewusst zu gestalten und zu steuern.

Aber Innovationen garantieren nicht automatisch den Erfolg, sondern sie bergen auch Risiken. Von besonderer Wichtigkeit ist auch der Begriff der Innovationsgeschwindigkeit (Time-to-Market), um möglichst zeiteffizient die Innovation realisieren zu können, wobei Flexibilität, technische Perfektion und sicheres Qualitätsniveau als Eckpfeiler zur Schaffung von Wettbewerbsvorteilen ebenfalls hohe Priorität besitzen. Gerade aus diesen Gründen gebührt dem Controlling im Bereich der Innovation eine besondere Bedeutung.[20]

Der F&E-Prozeß beginnt zwar mit der Wertschöpfungskette, aber erst die folgenden konkreten Projekte werden physisch greifbar und im Sinne des Controlling messbar.

[19] Roman Boutellier: „Innovationscontrolling", Seite 7.
[20] Roman Boutellier: „Innovationscontrolling", Seite 92.

Aus Sicht des Controllings ist der Innovationsbereich grundsätzlich schwerer zu bewerten als z.B. die Produktion. Schwierig ist beispielweise die Messbarkeit der Leistung und die fehlende Zuordnung von Ergebnissen in Form von Erlösen.
Die Innovation erfordert eine stärkere interdisziplinäre Vernetzung von Controllingkennwerten aus unterschiedlichen Unternehmensbereichen. Vieles lässt sich nicht in Zahlen erfassen. Für die Kreativität und die Entstehung neuer Ideen muss das Innovationscontrolling einen Freiraum lassen. So steht das Controlling vor der Herausforderung, einen Mittelweg zwischen den organisatorischen Gestaltungsmaßnahmen der Innovationsförderung und der Innovationsüberwachung zu finden. Somit ist die zentrale Aufgabe des Innovationscontrollings die Schaffung von Ergebnistransparenz entlang der gesamten Innovationswertschöpfungskette.
Das Innovationscontrolling beschäftigt sich mit der Frage, welche Projekte grundsätzlich benötigt werden, und es erarbeitet ein Soll-Portfolio der Projekte. Danach wird das Portfolio aufgrund verschiedener Kriterien analysiert. Im Anschluss wird entschieden, welche Projekte realisiert werden.[21]

3.2.4 Projektcontrolling

Sehr eng mit dem Innovationscontrolling ist das im Zeitablauf folgende Projektcontrolling verbunden (vgl. Abbildung 2: Ausbaustufen des F&E-Controllings). Der Übergang zwischen Innovationscontrolling und Projektcontrolling kann an der Schnittstelle fliessend sein, weil sich beide mit Projektzielen befassen und solche festlegen.
Die beim Innovationscontrolling erarbeiteten Projektziele bilden die Zielgrösse des Projektcontrollings. Diese Ziele werden übernommen und in Teil- oder Unterziele untergliedert.
Aufgaben des Projektcontrollings sind die Projektplanung, die Projektsteuerung und die Projektkontrolle sowie der Durchführung von Wirtschaftlichkeitsanalysen.

[21] Vgl:Boutellier, Völker, Voit: "Innovationscontrolling", Seite 5 f.

Bei der Planung des Projekts werden die einzelnen Phasen zuerst grob definiert, während in der Detailplanung die jeweils nächste Phase genauer bestimmt wird. Im Verlauf des Projektes kommt es ständig zu neuen Erkenntnissen und Situationen, so dass die Pläne regelmäßig aktualisiert werden müssen. In diesem Sinne ist die Planung als ein projektbegleitender, dynamischer Prozess zu verstehen. Zur Durchführung stehen verschiedene Instrumente zur Verfügung. Häufig werden Netzpläne, Balkendiagramme und Einsatzmittel-Auslastungsdiagramme verwendet.

Bezugsobjekte sind dabei vor allem Projektfortschritt (Meilensteine), Termine, Kapazitäten, Projektkosten, Qualität und Wirtschaftlichkeit. Zur Ermittlung des Projektfortschritts hat sich eine regelmäßige Berichterstattung der Mitarbeiter, die vom Controller ausgewertet und dokumentiert wird, als zweckmäßig erwiesen. Die Projektkosten können mit Hilfe einer prozessorientierten Projektkalkulation überwacht werden.

Eine wichtige Aufgabe des Projektcontrollings ist der Aufbau einer Erfahrungsdatenbank. Dadurch kann sich das Controlling auf Erfahrungswerte stützen, welche insbesondere bei der Planungsunterstützung des Managements sehr wertvoll sein können. Die Daten können als Basis für Aufwandschätzungen verwendet werden.

Das Berichtswesen für das Führungssystem stammt ebenfalls aus einer solchen Datenbank. Wirtschaftlichkeitsanalysen sollten projektbegleitend durchgeführt werden und können im Rahmen einer Rückbetrachtung wichtige Daten für die Erfahrungsdatenbank liefern (Kennzahlen).[22]

[22] Vgl. Horvath, “Controlling”, Seite 708 f.

3.2.5 Werkzeuge des Controllings

Instrumente des strategischen Controllings

Das strategische Controlling ist stark umwelt- und marktorientiert. Hier bedient man sich Analysetechniken, wie der Umwelt-, Markt-, Konkurrenz-, und der Unternehmensanalyse.[23]

Stärken-Schwächen-Analyse

Die Stärken-Schwächen-Analyse ist auch als Potentialanalyse bekannt. Die Potentialanalyse vergleicht die Stärken und Schwächen des eigenen Unternehmens mit denen des wichtigsten Wettbewerbers und zeigt die noch ungenutzten Potentiale auf. Das Potential ist eine immaterielle Größe, die für den späteren Gewinn eines Unternehmens eine wichtige Rolle spielt. Es stellt die zukünftigen Möglichkeiten dar, die geplant, beobachtet, analysiert und gesteuert werden müssen. Unter Stärken versteht man die Chancen der eigenen Unternehmung in der Zukunft und die Ursachen für vergangene Erfolge.[24]

Technologieanalyse

Die Technologieanalyse soll gewährleisten, dass von verschiedenen technischen Verfahren oder Systemen, die gewählt werden, die eine ausreichend gute Anpassung an die betrieblichen Verhältnisse zulassen, die den technischen Zielanforderungen genügen und die hinsichtlich der Kosten optimal arbeiten.

Planungssysteme

Planungssysteme kann man in Ablaufsysteme und Ereignissysteme unterscheiden.

[23] Vgl. Ziegenbein: „Controlling“, Seite 64.
[24] Vgl. Horvath: „Controlling”, Seite 370.

Ablaufsysteme sind z.B. Netzplantechniken. Man soll sicherstellen, dass alle notwendigen Handlungen in der richtigen Reihenfolge, zum richtigen Zeitpunkt und mit der richtigen Zeitdauer geplant werden.
Ein Ereignissystem ist z.B. die integrierte Unternehmensplanung, bei der organisatorische Einheiten hinsichtlich ihrer Leistungen, ihrer Faktoreinsätze und der davon abhängigen Kosten über einen Zeitraum geplant werden, um dadurch das Unternehmensergebnis vorhersagen zu können. Man unterstellt hierbei, dass ein bestimmtes Ereignisbündel eintritt.
Das Planungssystem soll dabei helfen, die Auswirkungen unterschiedlicher Ereignisbündel oder Annahmenkataloge bei sonst unveränderten Bedingungen aufzuzeigen.[25]

Kennzahlensysteme

Kennzahlen sind Beziehungsgrößen, die bei Überschreitung, Unterschreitung oder Abweichung vorgegebener Sollgrößen zu Aufmerksamkeit führen. Die Erstellung von Kennzahlen ohne bestimmte Zweckverfolgung ist nicht sinnvoll. Kennzahlen unterscheiden sich häufig auch inhaltlich von Unternehmen zu Unternehmen. Wenn man ein Kennzahlensystem aufbaut, sollte die Interpretation vorher den Informationsempfängern intensiv bekannt gemacht werden. Es empfiehlt sich, kritische Grenzwerte für häufig verwendete Kennzahlen zu ermitteln.[26]

Checklisten-Technik

Ergänzend zu Zahlenanalysen sind Ablauf- und Maßnahmenanalysen durchzuführen, um festzustellen, durch welche Handlungsweisen es zu den ermittelten Zahlen gekommen ist. Methoden hierfür sind Fragenkataloge oder Checklisten. Die Checklisten dienen nur der eigenen Vollständigkeitskontrolle der zu stellenden Fragen bzw. der zu ermittelnden Informationsdetails.

[25] Vgl. Baum,Coenenberg, Günther: „Strategisches Controlling”, Seite 309.
[26] Vgl. Boutellier, Völker, Voit: "Innovationscontrolling", Seite 5 f

Nutzwertanalyse

Die Nutzwertanalyse ist die Analyse einer Menge komplexer Handlungsalternativen mit dem Ziel, die Elemente dieser Menge entsprechend den Präferenzen des Entscheidungsträgers bezüglich eines Zielsystems zu ordnen.[27]

Schritt 1: Aufstellung eines Zielprogramms mit einzelnen Zielkriterien

Schritt 2: Bewertung der Alternativen hinsichtlich ihrer Erfüllung eines jeden Zielkriteriums in einer Bewertungsmatrix

Schritt 3: Gewichtung der Zielkriterien hinsichtlich ihres subjektiven Nutzens

Schritt 4: Rangordnung der Nutzwerte

Genehmigte Projektvorschläge können mit Hilfe der Nutzwertanalyse in eine Rangfolge gebracht werden. Hierzu werden Kriterien genutzt, um die Attraktivität eines Projektes einerseits und das Potential eines Projektes andererseits zu bewerten. Diese Kriterien werden als graduelle Kriterien bezeichnet und mit Hilfe einer Skala bewertet. Anhand einer gewichteten Berechnung ergibt sich der Gesamtnutzwert des Projektes, die Projektpunktzahl. Diese zeigt die Tendenz für das Projekt auf.[28]

Aus den Ergebnissen der Nutzwertanalyse erfolgen die Daten für die Portfolioanalyse.

[27] Vgl. Horvath, „Controling“, Seiten 507.

[28] Vgl:Boutellier, Völker, Voit: "Innovationscontrolling", Seite 164.

Portfolio-Analyse

Die Darstellung von Projekten im Rahmen von Portfolios dient aufgrund der Menge an Projekten und Projektideen als sinnvolle Entscheidungsgrundlage. Um die verschiedenen Aspekte eines Projektes transparent zu machen, wird auf Portfoliokonzepte zurückgegriffen.[29]

Dazu gehören:

- Risikoportfolio
- Wirtschaftlichkeits - Analyse
- Meilenstein-Trend - Analyse
- Projekt-Status - Analyse
- Kosten-Trend – Analyse

Projektideen können mit Hilfe der Portfolio-Analyse in bestimmte Aspekte eingeteilt werden. In Abbildung 3: wird eine Portfolioanalyse dargaestellt um die die Projekte nach wirtschaftlicher oder wissenschaftlicher Attraktivität zu ordnen.

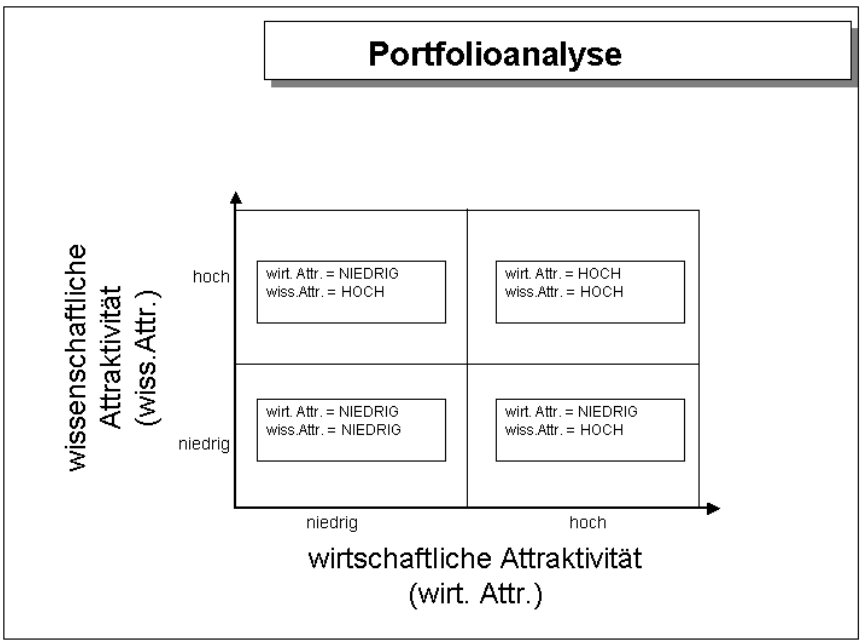


Abbildung 3: Portfolioanalyse

[29] Vgl:Boutellier, Völker, Voit: "Innovationscontrolling", Seite 12.

3.3 Projektbeurteilung und Projektplanung

Das Innovationscontrolling für die Auswahl der Projektideen in der Stiftung caesar ist wie folgt geplant:[30]

3.3.1 Von der Projektidee zur langfristigen Finanzierung der Forschung

Die Durchführung eines wissenschaftlichen Projektes als Gegenstand der Stiftungsarbeit bei caesar gliedert sich in folgende Phasen:

1. Projektvorlauf
2. Projektplanung
3. Projektstart / Projektdurchführung
4. Projektabschluss / Abrechnung
5. Folgephase

In jeder Phase ergeben sich unterschiedliche Anforderungen an das EDV – Konzept der Stiftung caesar. Im folgenden werden die Inhalte der Phasen beschrieben und die konzeptionelle Umsetzung der Anforderungen dargestellt.

3.3.1.1 Projektvorlauf

In dieser Phase werden die Ideen für neue Projekte gesammelt und bewertet. Sie gliedern sich wie folgt:

Erstkontakt mit der Projektidee
Folgekontakte (bei Interesse)
Vorselektionsentscheidung
Projektantrag
Projektgutachten
Projektbeschluss

[30] Vgl. Wagner: „Wissenschaftliche Projekte“, Stiftung caesar.

Gemäß dem Prinzip der Vorgangsdokumentation werden alle Aktivitäten in dieser Phase erfasst und stehen für das Berichtswesen zur Verfügung.
Erstkontakt / Folgedokumentation:
Die ersten Kontakte mit einer Projektidee vollziehen sich formlos. Sie sind zu dokumentieren und in einer Vorakte (Papier oder elektrisches Archiv) abzulegen.

Vorselektion:
Aus dem Pool der Ideen werden zu gegebenem Zeitpunkt besonders attraktive Projekte zur näheren Prüfung ausgewählt. An dieser Stelle geht der informelle Charakter der Kontakte über in einen Prozess, der insbesondere im Projektantrag und Projektgutachten eine besondere konzeptionelle Würdigung erfahren muss.

Projektantrag:
Ein vollständiger Projektantrag gliedert sich in zwei Teile, nämlich:

1. Projektbeschreibung
2. Detailplanung

Projektbeschreibung:
In der Projektbeschreibung sollten alle wesentlichen Aspekte des Vorhabens explizit verbal dokumentiert werden. Im Zusammenhang mit einem Forschungsprojekt ist ein weitgefächertes Spektrum von Vorüberlegungen denkbar. Im folgenden sind einige dieser Aspekte aufgelistet. Für den Antragsteller sollte eine Checkliste entwickelt werden, in der die von der Geschäftsleitung als wesentlich erachteten Kriterien zusammengestellt sind. Unter Einbeziehung dieser Checkliste kann bei Bedarf ein Projektantragsformular erstellt werden.

Detailplanung:

Projektskizzierung:

- Wissenschaftliche Disziplin
- Forschungsfeld
- Problemstellung und derzeitige Lage (Stand der Forschung)
- Ziel des Projektes
- Vorgehensmethodik
- Ablauf
- Mögliche Ergebnisse
- Internationalität
- Geplante Kooperationen
- Regionalaspekt
- Welche Beiträge wird das Projekt zur Erreichung der Ziele von caesar leisten?

Projektteam / (Triplet):

- Konzept für Zusammenstellung
- Verfügbarkeit geeigneter Kandidaten
- Perspektiven und Pläne nach Projektabschluss

Technische Merkmale:

- Beschreibung eingesetzter Technologie, hinsichtlich Investitionsvolumen und Verfügbarkeit.
- Fallen Anschaffungskosten an?
- Kann alternativ auf fremde Ressourcen zurückgegriffen werden?
- Verwertbarkeit
- Alternative Nutzungsmöglichkeiten nach Projektende
- Innovationsgeschwindigkeit
- Sicherheitsrisiken und Vorgaben
- Materialintensität
- Logistischer Rahmen
- Raumbedarf

- Gesetzliche Bestimmungen
- Bestehen Patente, die den Forschungsprozess erschweren?

Projektablauf:

- Terminplan
- Meilensteine im Projektablauf
- Ist Einbeziehung von Fremdforschung geplant?
- Ist Fremdbezug bei Terminengpass möglich?
- Geplante Kooperationen

Projektrisiken:

- Forschungs- / Fehlschlagsrisiko in Planperiode
- Identifikation von möglichen Problemen
- Lösungsansätze für identifizierte Probleme
- Konkurrenz in der Forschung durch andere Institute / Firmen (National / International)
- Innovationsgeschwindigkeit (Zeitdruck)
- Wirtschaftliche Risiken
- Absatzrisiken
- Änderung der Nachfrage

Projektgutachten:

Das Projektgutachten gliedert sich in eine wissenschaftliche Vorentscheidung und in eine betriebswirtschaftliche Gesamtbewertung. Grundlage für das Gutachten bilden die Angaben des Projektantrages.

Für die eigentliche Begutachtung sind folgende Kriterien denkbar:

- Wissenschaftliche Attraktivität
- Beurteilung wirtschaftlichen Potentials
- Risikobeurteilung
- Gutachten zur eigenen Projektstärke
- Abschließende Kommentierung betriebswirtschaftlicher Aspekte
- Durchführbarkeits-Überprüfung
- Portfoliobildung

Der Projektantrag und das Projektgutachten müssen im System abgebildet werden, um eine Historie aufbauen zu können.

3.3.1.2 Projektbeschluss

Basierend auf dem Ergebnis des Gutachtens erfolgt ein Projektbeschluss. Neben der Gesamtprojektbeurteilung ist die Frage zu klären, ob das Projekt finanzierbar ist. Vor dem Hintergrund knapper finanzieller Ressourcen ist für die Beschlussfassung auch der Vergleich des Antrages mit weiteren Alternativprojekten unabdingbar. Hier ergeben sich gewisse Analogien zur Ermittlung eines optimalen Investitionsprogramms.
Entscheidet sich caesar für ein Projekt, so bilden die in Projektantrag und Gutachten gesammelten Daten die Grundlage für die anschließende Detailplanung des Projektes und für die Zielvorgaben im Innovationscontrolling.

3.3.1.3 Projektplanung

Für die Planung können die geplanten Tools aus Antrag und Gutachten verwendet werden. Als zusätzliches Element ist ein Feinplanungstool für den Projektablauf einzuführen, der sich erst in dieser Phase konkretisiert.

3.3.1.4 Projektstart / Projektdurchführung

Die betriebswirtschaftliche Seite der Projekte wird durch das kaufmännische Buchhaltungssystem (*Oracle Financials*) abgebildet.
Statistische und wissenschaftliche Belange werden zudem anhand des Controlling-Konzeptes von caesar abgebildet. Die Daten hierfür sind durch die Vorgangerfassung im System vorhanden.

3.3.1.5 Projektabschluss

Nach Abschluss des Projektes erfolgt eine abschließende wissenschaftliche sowie eine betriebswirtschaftliche Gesamtbeurteilung des Projektes unter Einbeziehung der vollständigen Projekthistorie. Auch eine Aktualisierung der Verwertungslage, Marktanalyse und Absatzprognose sollten im kaufmännische Buchhaltungssystem (*Oracle Financials*) stattfinden. Die beschriebenen Prozesse sollten turnusmäßig auch im laufenden Projekt erfolgen, um Aufschlüsse über die Sinnhaltigkeit der Projektfortführung zu erhalten.

3.3.2 Folgephase

Auch nach Abschluss des eigentlichen Projektes ergeben sich aus der Verwertung der wirtschaftlichen und wissenschaftlichen Ergebnisse ständig neue Aspekte. Daher sollte das Projekt offengehalten werden.
Eine Angliederung in den Vertrieb bzw. an die Dokumentationsstelle ist denkbar. Zu den Aufgaben der entsprechenden Stelle gehört unter anderem die Verfolgung und Begutachtung der wirtschaftlichen Potentiale und deren fortlaufende Planung.
Die gesammelten Informationen sind für das Controlling der Gesamtstiftungsarbeit notwendig. Zudem liefern die Umsätze in dieser Reifephase einen sehr wichtigen Beitrag zur langfristigen Finanzierung des Stiftungsgeschäfts und bilden die Grundlage für die Abrechnung der Erfindervergütungen.

4. Datenbankentwicklung

Wie im vorigen Kapitel über Innovationscontrolling dargestellt, wird in der Stiftung caesar eine Datenbankanwendung benötigt, in der Projektideen erfasst und bewertet werden können. In diesem Abschnitt wird die Datenbankentwicklung am *Beispiel InnoCon* beschrieben.

4.1 Phasenmodell der Datenbankentwicklung

Datenbankentwicklungen können in bestimmte Phasen eingeteilt werden (vgl. Abbildung 4: Phasenmodell der Datenbankentwicklung):

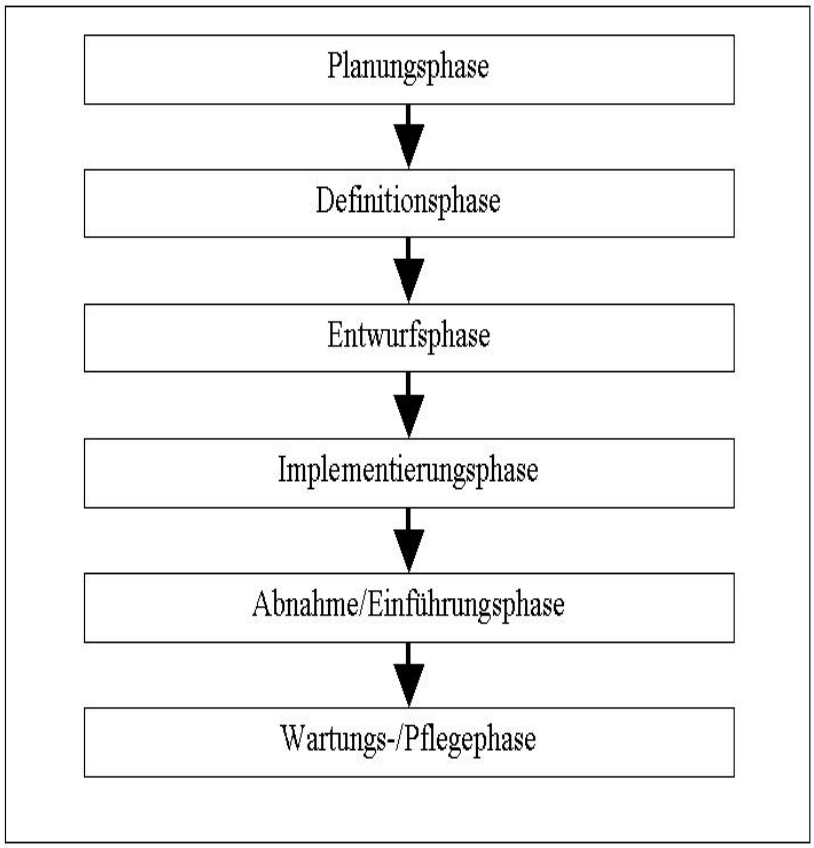


Abbildung 4: Phasenmodell der Datenbankentwicklung[31]

[31] Vgl. Balzert: Lehrbuch der Softwaretechnik, zitiert nach: Faeskorn-Woyke: Vorlesungsscript „Datenbanksysteme“ (2000) Seite 46, Abb. 2.1.

Dieses Phasenmodell lässt sich wie folgt beschreiben:

„In der **Planungsphase** werden die Systemfunktionen und Daten in groben Zügen geplant und festgelegt.
Die **Definitionsphase** ist eine Verfeinerung der Planungsphase, in der eine explizite Systemdefinition in Form eines Pflichtenheftes vorgenommen wird. Das geschieht noch umgangssprachlich in einer auch für Laien lesbaren Form.
In der **Entwurfsphase** wird das Ergebnis der Definitionsphase mit Mitteln der Softwaretechnik, wie dem Entity-Relationship-Modell, genau beschrieben und für die **Implementierung** vorbereitet.
In der **Abnahme und Einführungsphase** werden häufig Daten aus Altsystemen übernommen und das fertige System auf den Ernstfall vorbereitet und dann auch gestartet. In der **Wartungs- und Pflegephase** fallen oft zusätzliche Erweiterungen an oder es werden noch Fehler, die sich erst zur Laufzeit eines Systems herausstellten, behoben.“ [32]

4.2 Definitionsphase (Pflichtenheft)

In der Definitionsphase werden zunächst die Anforderungen, die der Nutzer an die zu entwerfenden Datenbank stellt, ermittelt. Dies kann mittels Interviews mit den einzelnen Benutzern oder mit Hilfe von Fragebögen geschehen. Das Ergebnisdokument der Definitionsphase wird in einem Anforderungs-Dokument, dem sogenannten Pflichtenheft, schriftlich festgelegt. Dieses Dokument muss von den Nutzern angenommen werden und bildet dann die Grundlage für den folgenden Entwurf.
In dem Pflichtenheft werden die Funktionen und Daten des Systems aufgeführt.

Die folgenden Ausführungen lehnen sich an das Vorlesungsskript „Datenbank-systeme“ von Frau Prof. Faeskorn-Woyke an.

[32] Faeskorn-Woyke: Vorlesungsskript „Datenbanksysteme“ (2000) Seite 46.

Pflichtenheft der Anwendung InnoCon:

4.2.1 Zielbestimmung

Ziel ist die Erstellung eines Innovationscontrolling-Instrumentes, das als Modul in *Oracle Applications* realisiert werden soll.

Der Zweck dieses Moduls ist, für den wissenschaftlichen Vorstand von *caesar* eine Datenbank-Anwendung zu erstellen, in der Projektideen erfasst und bewertet werden können. Um außerhalb der Stiftung, z.B. auf Dienstreisen, auf die Datenbankanwendung zugreifen zu können, muss diese per Internet zur Verfügung gestellt werden. Zusätzlich sollen Auswertungstools geschaffen werden, aus denen ersichtlich wird, wie die Projekte von der Erstdokumentation bis zur Fertigstellung verlaufen.

Die Entscheidung, ob eine Projektidee realisiert werden soll, trifft der wissenschaftliche Leiter der Stiftung caesar. Dazu werden Entscheidungshilfen in Form von verschiedenen Berichten (Reports) erstellt.

4.2.2 Musskriterien

Der Vorgang für das Modul ist wie folgt geplant:

1. Erstdokumentation (Erfassung der Projektidee):

   Die Projektidee wird von demjenigen formuliert, der die entsprechende Idee hat. Der Vorschlag, oder auch die Erstdokumentation, beginnt mit dem Titel (Kurzbeschreibung) und dem Ersterfassungs-Datum sowie der Beschreibung des angedachten Projektes. In der Projektbeschreibung sollten der Projektablauf, die Kosten und die Termine sowie die Teammitglieder, falls diese nicht extern hinzugezogen werden müssen, vorgeschlagen werden. Zusätzlich sollen in der Erstdokumentation die zugehörigen Bereiche wie Forschungsbereich, Triplet, Projektpotential und die Kontakte, die zu dieser Projektidee führten, erfasst werden. Nach der Erfassung kann die Projektidee einem Vorgesetzten oder einem anderen Mitarbeiter als Wiedervorlage vorgelegt werden.

2. Folgedokumente oder auch Folgekontakte (bei Interesse):

   Nachdem die Projektidee erfasst wurde, sollen Folgedokumente oder geführte Gespräche erfasst werden, um Unklarheiten auszuräumen und um das Projekt zu spezifizieren.

3. Vorselektion der Projektidee:

   In der Vorselektion wird entschieden, ob die Projektidee *weiterverfolgt*, *zurückgestellt* oder *nicht weiterverfolgt* werden soll. Jedweder Entscheidung muss eine entsprechende Begründung beigefügt sein.

4. Berichte (Reports) :

   Berichte dienen als Entscheidungshilfen für die Realisierung der Projektidee. Zu diesem Zweck sind folgende Berichte zu erstellen:

   a) Erstdokumentation:

      Ausgabe der erfassten Projektidee

   b) Vorselektion:

      Ausgabe der Projektidee und des Vorselektions-Beschlusses sowie der Begründung für die Entscheidung

   c) Folgedokumente:

      Ausgabe der Projektidee sowie der zugehörigen Folgedokumente

   d) Projektvorlauf:

      Ausgabe der Vorlaufphasen der einzelnen Projektideen

   e) Projektverlaufszeitanalyse:

      Ausgabe über den Gesamtablauf der Projektphasen (Vorselektionsentscheidung) sowie den zeitlichen Abstand von der Ersterfassung der Idee bis zum Projektabschluss

4.2.3 Wunschkriterien

a) Anfügen von Dokumenten:

Nachdem der Entschluss für die Realisierung der Projektidee in Form der Vorselektionsentscheidung feststeht, sollten der Projektidee der Projektantrag, das Projektgutachten sowie der Projektbeschluss in Form von verschiedenen Dateiformaten wie z.B. Word-Dokumente beigefügt werden können. Diese Dokumente sollten dann in der Datenbank gespeichert werden.

b) Weiterleiten der Projektidee:

Die erfasste Projektidee sollte an andere Mitarbeiter der Stiftung caesar weitergeleitet werden können.

4.2.4 Abgrenzungskriterien

Der Zugriff von externen Einrichtungen (z.B. Universitäten, externe Forscher) ist nicht Bestandteil dieses Projektes.

4.2.5 Produkteinsatz

Das Produkt wird in der Stiftung caesar eingesetzt. Die Zielgruppen des Produktes sind der wissenschaftliche Vorstand sowie sämtliche Mitarbeiter der Stiftung caesar. Die Anwendung muss jederzeit im Intranet verfügbar sein.

4.2.6 Produktumgebung

Softwarevoraussetzungen:

Anwenderseite: Javafähiger Internetbrowser: Netscape Navigator 4.6, sowie Oracle JInitiator 1.1.7.

Serverseite:

Folgende Software wird vorausgesetzt:

Datenbank Server Oracle 8.0.5 OS Sun 5.6
Concurrent Manager / Reports Server
Oracle Forms 4.5x Server

Oracle Web Application Server 3.0.2

Oracle Applications 11.0.300 NLS D

Hardwarevoraussetzungen:

Auf der Anwenderseite reicht ein minimal konfigurierter PC für den Internet-Betrieb.

Server:

Wie in der folgenden Tabelle aufgeführt, wird für den Server ein Festplattenbedarf von 31 Gigabyte benötigt:

Software	Platzbedarf in GB
Oracle Software (Server-DB, Tools, Forms, Reports)	3 GB
Produktions-Instanz:	
Applications-Software	4 GB
Datenbank	20 GB
Redolog Dateien	1 GB
Patches	2 GB
/tmp	1 GB
Summe	**31 GB**

Hauptspeicher und CPU

Entscheidend für die Berechnung des Hauptspeichers des Datenbankservers ist die Anzahl der Benutzer. Ausgehend von 100 Benutzern ergibt sich folgender Speicherbedarf:

	Speicher
100 User à 15 MB	1500 MB
SGA	200 MB
Datenbank Prozesse	150 MB
Concurrent Managers	150 MB
Betriebssystem incl. Filesystem Cache	200 MB

Daraus ergibt sich ein Gesamtspeicherbedarf von mindestens 2,2 Gigabyte für die Umgebung. Der Server sollte mit mindestens vier Prozessoren ausgestattet sein. Die Taktung sollte bei ungefähr 250 MHz liegen.

4.2.6.1 Produktfunktionen

a) Erfassung von Projektideen, Folgedokumenten sowie des Vorselektions-beschlusses
b) Suche nach bestimmten Projektideen
c) Ausgabe der erfassten Projektideen, der Folgedokumente und des Vorselektionsbeschlusses in Form von Reports

4.2.6.2 Produktdaten

a) Erstdokumentation:

Projekt (Projektnummer, Titel, Datum der Ersterfassung, Beschreibung, Projektart)
Kontaktperson (Anschrift)
Kontakte (Kontaktart, Kontaktgruppe)
Projektpotential (Gewichtung für verschiedene Bereiche)
Forschungsbereich (Disziplin, Themenkreis, Bereich)

b) Folgedokumentation:

(Datum, Gesprächsnotiz, vereinbarte Aktivitäten)

c) Vorselektion:

(Datum, Vorselektionsbeschluss, Begründung, zu veranlassende Aktivitäten)

d) Wiedervorlage:

(Datum, Verteiler an, Anlagen, Bearbeiter)

4.2.6.3 Benutzeroberfläche

- Einbindung in Oracle Applications
- Mausbedienung
- Menüorientierte Bedienung

4.2.6.4 Globale Testfälle

Alle Funktionen, Masken sowie Reports sind von den Nutzern zu testen.

4.2.6.5 Entwicklungsumgebungen

Für die Anwendung InnoCon werden folgende Entwicklungswerkzeuge benötigt:

- Designer/2000 Release 2.1
- Developer/2000 Release 4.5 für die Erstellung der Masken und Berichte
- SQL - Programm
- Telnet
- FTP-Programm
- Oracle Applications

4.3 Entwurfsphase

Der Entwurf einer Datenbank gliedert sich in folgende Phasen:

1. Anforderungsanalyse
2. konzeptioneller Entwurf
3. logischer Entwurf
4. physischer Entwurf
5. Implementierung

4.4 Konzeptioneller Entwurf (ER–Modellierung)

Aus der Anforderungsanalyse, dem Pflichtenheft, wird ein konzeptioneller Entwurf abgeleitet. In dem konzeptionellen Entwurf wird der Ausschnitt der realen Welt, der in der Datenbank gespeichert werden soll, in der graphischen Form eines Entity-Relationship-Modells (ER-Modells) dargestellt.
Das von P.P. Chen 1976 vorgestellte ER-Modell gilt als Standard-Modellierungstechnik im Datenbank-Bereich. Es existierten zahlreiche Erweiterungen zum ursprünglichen Ansatz sowie verschiedene graphische Darstellungsformen. Der Autor der vorliegenden Diplom-Arbeit hat sich für die Darstellungsform des Oracle Designers/2000 entschieden.
Das ER-Modell setzt sich aus wenigen Elementen zusammen, die zunächst einzeln beschrieben werden. Die Hauptbestandteile sind die Entitäten mit ihren Attributen und die Beziehungen (Relationships) zwischen den Entitäten.

4.4.1 Entitäten

Eine Entität beschreibt ein Objekt aus der realen Welt. Jede Entität besitzt einen Namen. Der Name wird durch ein Substantiv gebildet, das möglichst allgemein die Bedeutung der Entität beschreibt.
Im ER-Diagramm wird die Entität durch ein Rechteck mit abgerundeten Ecken dargestellt. (vgl. Abbildung 5: Beispiel für Entitäts-Typen und Attribute).
Jedes Exemplar einer Entität muss eindeutig durch ein Attribut oder eine Kombination von Attributen identifizierbar sein. Das heißt, dass ein bestimmtes Exemplar einer Entität nicht mehrfach vorkommen darf.

4.4.2 Attribute

Die Beschreibung von Entitäten erfolgt durch Attribute. Es gibt sogenannte Pflichtattribute, Kandidatenschlüssel (candidate key). Ein Kandidatenschlüssel kann von einem oder mehreren Attributen gebildet werden. Somit ist es möglich, ein Exemplar einer Entität eindeutig zu identifizieren. Wenn mehrere Kandidatenschlüssel identifiziert werden können, kann einer von Ihnen als Primärschlüssel ausgewählt werden.

Ein Primärschlüssel besteht immer aus obligatorischen Feldern, anderenfalls wäre eine Identifikation nicht sichergestellt. Die den Schlüssel bildenden Felder werden Schlüsselattribute (key attributs) genannt.

Im ER-Modell wird ein Primärschlüssel mit einem '#', obligatorische (Pflichtattribute) mit einem '*' und optionale Attribute mit einem 'o' Zeichen (vgl. Abbildung 5:Beispiel für Entitätstypen und Attribute) versehen:

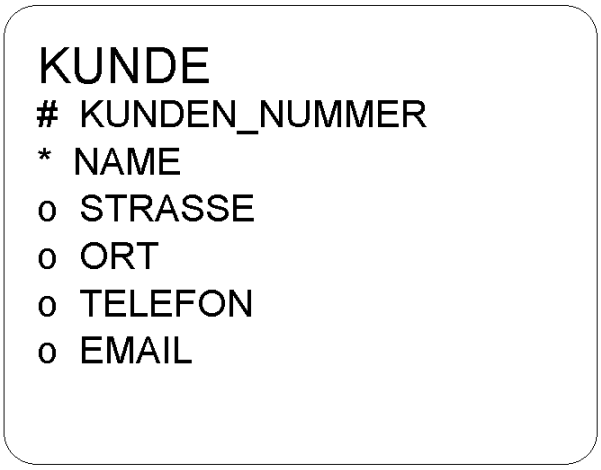


Abbildung 5: Beispiel für Entitäts-Typen und Attribute

4.4.3 Domänen

Eine Domäne ist ein Datentyp, der das Format und den Wertebereich von Attributen festlegt. Domänen werden eingesetzt, um gleichartige Attribute auf einer gemeinsamen Basis zu definieren und so eine Einheitlichkeit bei der Verwendung von Datentypen zu gewährleisten. Technisch gesehen sind Domänen abstrakte Datentypen.
Als Domäne kann beispielsweise folgendes eingesetzt werden:

Geschlecht: (männlich / weiblich)
Anrede: (Herr / Frau)
Aussagen: (ja / nein, wahr / falsch)

4.4.4 Beziehungen

Eine Beziehung stellt einen logischen Zusammenhang zwischen zwei Entitäten dar. Beide Endpunkte einer Beziehung haben einen Namen. Eine Beziehung legt fest, in welchem Verhältnis die Entitäten zueinander stehen. Dieses Verhältnis kann wie folgt beschrieben werden:

<u>Hierarchische Abhängigkeit:</u>
Ein bestimmtes Exemplar einer Entität kann nur existieren, wenn ein Exemplar einer übergeordneten Entität vorhanden ist.
<u>Übertragbare Beziehung:</u>
Die Beziehung von einem Exemplar einer Entität kann auf ein anderes wechseln.
<u>Kardinalität (Häufigkeit):</u>
Die Kardinalität legt fest, wie viele Exemplare einer anderen Entität zugeordnet werden können.
<u>Optionalität:</u>
Mit der Optionalität wird ausgedrückt, ob ein Exemplar einer anderen Entität zugeordnet werden muss oder nicht.[33]

[33] Vgl. R.Hoppe, M. Mempel „Oracle Designer R2.1“ Seite 100.

Beziehungen können einfach oder mehrfach (mit Nennung der Möglichkeiten), optional oder obligatorisch sein (vgl. Abbildung 6:: Beziehungsarten). Entitäten, die in keinem direkten Verhältnis zueinander stehen, werden nicht durch Relationen miteinander verbunden. Eine der am häufigsten benutzten Beziehungen ist die 1-N–Beziehung. Hierbei können Entitäten mehreren Exemplare einer zweiten (oder der gleichen) Entität zugeordnet werden.
Eine 1-C-Beziehung stellt eine Kann-Beziehung dar, wobei c für conditional steht.

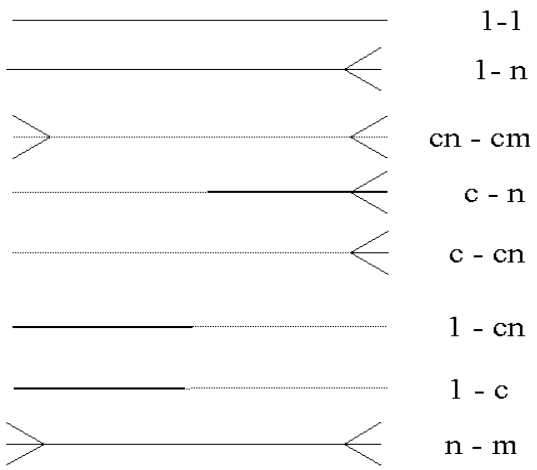
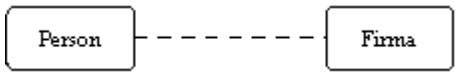


Abbildung 6: Beziehungsarten

Die Benennung erfolgt immer am Beginn und am Ende einer Beziehung. In Abhängigkeit von Kardinalität und Optionalität wird eine Beziehung immer als Satz gelesen. Um eine Beziehung vollständig beschreiben zu können, werden immer zwei Sätze gebildet.
Zum Beispiel lautet die Beziehung zwischen Person und Firma aus Sicht der Person wie folgt:

Eine Person **kann** bei **einer** Firma angestellt sein:

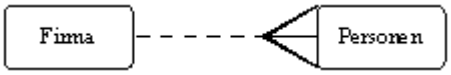
Darstellung im ER-Modell:



Andersherum aus Sicht der Firma lautet die Beziehung:

Eine Firma **kann mehrere** Personen beschäftigen:

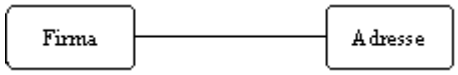
Darstellung im ER-Modell:



Eine obligatorische Beziehung wird entsprechend mit dem Hilfsverb <u>muss</u> gebildet:

Eine Firma **muss eine** Adresse haben:

Darstellung im ER-Modell:



4.5 Logischer Entwurf

Im logischen Entwurf erfolgt eine Umsetzung des in der vorherigen Phase erzeugten Datenmodells in das zu verwendende Datenbankschema. Meist wird hierbei das relationale Modell benutzt.

Aus dem Entity-Relationship-Modell werden also direkt die entsprechenden Relationentabellen generiert. Diese Tabellen müssen nun auf Einhaltung der Normalformen überprüft werden. Normalformen dienen der Vermeidung von Operationsanomalien, die zu Inkonsistenzen im Datenbankinhalt führen können. Relationentabellen, die nicht den Normalformen genügen, müssen entsprechend umgeformt werden.

4.5.1 Normalisierung der Relationen

Die entworfenen Relationen werden zur Vermeidung unerwünschter Abhängigkeiten, Eliminierung von Redundanzen (Wiederholung von abhängigen Daten) und zur besseren Verständlichkeit normalisiert. Hierfür werden die Daten mehrfach in Relationen aufgeteilt, so dass sie am Ende den Normalisierungsregeln entsprechen.

Die 1. Normalform

Die erste Normalform besteht einzig und allein aus der Forderung, für einzelne Attribute keine zusammengesetzten Datentypen zu erlauben.

Def.: „Eine Relation R ist in der ersten Normalform (1NF), wenn alle Attribute nur atomare Werte (keine zusammengesetzten Datentypen) enthalten.“[34]

Die 2. Normalform

Def.: „Eine Relation R mit Primärschlüssel S ist in der zweiten Normalform (2NF), wenn sie in der 1NF ist und jedes Nicht–Schlüssel–Attribut funktional abhängig vom Primärschlüssel S ist.“[35]

Funktionale Abhängigkeit

Ein Attribut A ist von Attribut B abhängig, wenn zu einem bestimmten Wert von A höchstens ein Wert von B möglich ist, d.h. immer, wenn A den Wert X hat, muss B den Wert Y haben.

Ein *Identifikationsschlüssel* (Schlüsselattribut) ist ein Attribut, von dem alle anderen Attribute einer Relation funktional abhängig sind.

Def.: **“**X und Y seien zwei Teilmengen von Attributen der Relation R.

Y heißt funktional abhängig von X, wenn folgendes gilt:
Für alle Tupel r, s aus R gilt:
Aus s[X] = r[X] folgt stets : s[Y] = r[Y]
Abkürzung: X -> Y
Y kann im mathematischen Sinne als Funktion von X interpretiert werden.
In einer Relation R = R(A, B) ist das Attribut B von dem Attribut A funktional abhängig, wenn zu jedem Wert des Attributs A genau ein Wert des Attributs B gehört.“[36]

[34] Faeskorn-Woyke: Vorlesungsskript „Datenbanksysteme“ (2000) Seite 125.
[35] Faeskorn-Woyke: Vorlesungsskript „Datenbanksysteme“ (2000) Seite 126.
[36] Faeskorn-Woyke: Vorlesungsskript „Datenbanksysteme“ (2000) Seite 129.

Die 3. Normalform

Def.: „Eine Relation R ist in der dritten Normalform (3NF), wenn sie sich in der 1NF und in der 2NF befindet und kein Sekundärattribut (Nicht–Schlüssel–Attribut) transitiv abhängig von einem Schlüsselattribut ist.“[37]

Transitive Abhängigkeit

Def.: „Mit X, Y und Z seien paarweise verschiedene Attributkombinationen einer Relation R = R (A1, A2, …, AN) bezeichnet.
Z heißt transitiv abhängig von X, wenn Y funktional abhängig von X und Z funktional abhängig von Y ist, aber X nicht funktional abhängig von Y ist.
In Zeichen: $X \rightarrow Y \rightarrow Z$ “[38]

[37] Faeskorn-Woyke: Vorlesungsskript „Datenbanksysteme“ (2000) Seite 129.
[38] Faeskorn-Woyke: Vorlesungsskript „Datenbanksysteme“ (2000) Seite 129.

4.6 Physischer Entwurf

Im physischen Entwurf wird entsprechend zu den generierten Relationentabellen der Datenbankcode für das zu verwendende Datenbanksystem erzeugt.

4.7 Implementierungsphase

Die Implementierungsphase besteht aus der tatsächlichen Implementierung und der Installation der Datenbank auf dem verwendeten Datenbanksystem.

4.8 Abnahme und Einführungsphase

In der Abnahmephase wird die Datenbankanwendung den Benutzern vorgestellt. Bei komplizierten Anwendungen können Schulungen durchgeführt werden.

4.9 Wartungs- und Pflegephase

Jede Datenbankentwicklung benötigt Wartungsarbeiten. Dazu gehört das Prüfen des Speicherplatzes der Tabellen, der Datenbank und des Servers, in dem die Datenbankanwendung implementiert wurde. Diese Arbeiten werden von einem Systemverwalter erledigt, der auch für die Erweiterbarkeit der Anwendung zuständig sein kann.

5. Entwicklungswerkzeuge

In diesem Abschnitt werden die benutzten Entwicklungswerkzeuge, mit denen die Datenbankanwendung InnoCon realisiert wurde, vorgestellt. Dazu gehören der *Oracle Designer/2000* sowie der *Oracle Developer/2000*. Das ER-Modell wurde mit Hilfe des *Oracle Designers/2000*, die Masken und die Berichte (Reports) mit dem Tool *Developer/2000* erstellt. Die erstellten Masken und Berichte wurden in *Oracle Applications* eingebunden, dazu wird im folgenden Kapitel auf *Oracle Applications* eingegangen. Zuvor wird die Sprache SQL beschrieben.

5.1 SQL

SQL (Structured Query Language) ist eine standardisierte Abfragesprache, die alle Sprachelemente enthält, die erforderlich sind, um sämtliche Arbeiten, die beim Umgang mit einer relationalen Datenbank anfallen, auszuführen. SQL ist eine Programmiersprache der vierten Generation, eine sogenannte 4GL-Sprache.[39] Die Sprache SQL dient der Kommunikation mit einer relationalen Datenbank. Man kann damit Datenbanken abfragen und Tabellen in der Datenbank modifizieren. Mit SQL-Anweisungen weist man die Datenbank an, was zu tun ist, und nicht, wie es zu tun ist. In SQL unterscheidet man mehrere Befehlsgruppen:

[39] Vgl: Scott Urmann, „Oracle PL/SQL Programmierung“, Seite 1.

Data Definition Language (DDL)

Die Befehlsgruppe DDL dient zur Datendefinition. Dazu gehören Befehle zum Aufbau von Tabellen: z.B. CREATE (erstellen), ALTER (ändern) und DROP (löschen) von Table, View, Index.

Data Manipulation Language (DML)

DML umfasst alle Anweisungen, die Daten erzeugen, verändern, löschen oder auswerten, z.B. INSERT, APPEND, UPDATE, SELECT.

Data Control Language (DCL)

Darüber hinaus gibt es Befehlsgruppen, die die Vergabe von Zugriffsrechten steuern und die Struktur und die Größe der Datenbank beeinflussen.

Manipulierte Objekte in SQL:

- Table: Eine Tabelle ist eine Datenstruktur, die in einer relationalen Datenbank die Daten enthält. Sie besteht aus Zeilen (Column) und Spalten (Row).

- Column: Jede Spalte repräsentiert ein Attribut der Tabelle.

- Row: Zeilen speichern die Daten einer Tabelle.

- Value: Dies ist ein Wert beispielsweise ein Datum, der durch den Schnitt einer gegebenen Zeile und Spalte referenziert wird. Ein Wert nimmt den Datentyp seiner Spalte an.

- View: Eine View ist die logische Repräsentation einer Tabelle oder einer Kombination von Tabellen. Eine View erhält ihre Daten aus den Tabellen.

- Index Indizes werden in Datenbanken für zwei verschiedene Zwecke benötigt: Beschleunigung des Zugriffs auf die Zeilen einer Tabelle und die Erzwingung der Eindeutigkeit der Zeilen einer Tabelle.

- Sequence: mit Sequenzen werden eindeutige Nummern vergeben. Auf eine vorhandene Sequenz kann durch *name_der_sequenz.CURRVAL* oder *name_der_sequenz.NEXTVAL* zugegriffen werden.
  CURRVAL liefert den aktuellen Wert während NEXTVAL den Wert der Sequenz um eins erhöht und den neuen Wert zurückgibt.

SQL - Abfrage

Die Grundstruktur einer SQL-Abfrage hat folgende syntaktische Form:

select <Attributliste>
from <Tabelle>
where <Bedingung> (Auswahlkriterium kann auch leer sein)

Um die Datensätze einer Tabelle zu zählen, wird der Befehl COUNT benutzt. Möchte man die Anzahl aller vorhandenen Kunden erfahren, wird folgende Abfrage gestellt:

select count (*) from kunden

Es existieren weitere SQL-Befehle, auf die hier nicht näher eingegangen werden kann. Alle SQL-Befehle können bei Oracle mit dem Programm SQL*Plus. ausgeführt werden.

5.1.1 PL/SQL

PL/SQL steht für Procedural Language/SQL und ist eine Erweiterung von SQL. In PL/SQL wird die Sprache SQL um folgende Elemente erweitert:

- Variablen und Typen
- Steuerstrukturen, wie IF-THEN-ELSE-Anweisung und Schleifen
- Prozeduren und Funktionen
- Objekttypen und Methoden[40]

Alle PL/SQL-Programme setzen sich aus Blöcken zusammen. Es gibt anonyme Blöcke und benannte Blöcke. Diese werden dynamisch erzeugt und nur einmal ausgeführt.

[40] Vgl: Scott Urmann, „Oracle PL/SQL Programmierung“, Seite 2.

Aufbau eines PL/SQL-Blocks:

```
DECLARE          -- Deklaration der Variablen, Typen
 BEGIN           -- Beginn der executable section, enthält PL-Anweisungen
                 -- und SQL-Statements

 EXCEPTION       -- Fehlerbehandlung
END;             -- Programmende
```

Unterprogramme sind Prozeduren (Procedure), Funktion (Function) oder Pakete (Package), die in einer Datenbank gespeichert werden. Nachdem Sie einmal erstellt wurden, verändern sich diese Blöcke nicht mehr.

Trigger sind benannte PL/SQL-Blöcke, die auch in der Datenbank gespeichert werden. Trigger werden immer dann ausgelöst, wenn das zu dem Trigger gehörende Ereignis auftritt. Dies kann z.B. vor oder nach DELETE, INSERT oder UPDATE der Tabelle der Fall sein. Das Löschen eines Triggers erfolgt mit DROP TRIGGER, das Ändern mit ALTER TRIGGER.

5.1.1.1 Pakete

Pakete (Package) sind Datenbankobjekte, in denen zur Logik gehörende PL/SQL-Arten, Objekte und Unterprogramme zusammengefasst sind. Pakete setzen sich in der Regel aus zwei Dateien zusammen: eine Datei mit der Paketspezifikation und eine Datei mit dem Paketrumpf.
Die Spezifikationsdatei stellt die Schnittstelle zu der Anwendung dar. In ihr sind die Arten, Variablen, Konstanten, Ausnahmen, Cursor und Unterprogramme angegeben, die im Paket zur Verfügung stehen. Die Spezifikationsdatei enthält die Bezeichnung des Pakets sowie die Erläuterungen der Funktionen von Prozeduren.
Der Paketrumpf definiert Cursor und Unterprogramme, dieser enthält die tatsächliche PL/SQL-Kennung für die Prozeduren und implementiert so die Spezifikation.

5.1.1.2 Prozeduren

Prozeduren (Procedure) sind Unterprogramme eines Pakets, die von der Anwendung aufgerufen werden und eine bestimmte Handlung ausführen. Prozeduren legen fest, welche Parameter als Kontext für das Programm übertragen werden, wie die Eingabewerte verarbeitet werden und welche Ausgabewerte abgerufen werden.

Jede Prozedur hat vordefinierte Eingabeparameter, die in der vordefinierten Reihenfolge übertragen werden müssen. Die Parameter kennzeichnen die gerade verarbeitete Transaktion sowie den Kontext, in dem das Programm aufgerufen wird.

Die Prozedur verwendet die Eingaben und führt logische Verarbeitungen und Berechnungen durch. Das Programm kann einfach strukturiert sein und z.B. eine Kundennummer ausgeben, oder aber einen komplexen Algorithmus darstellen, der eine Reihe von Funktionen ausführt.

Eine Prozedur setzt sich aus zwei Teilen zusammen, aus der Spezifikation und dem Rumpf. Die Spezifikation der Prozedur beginnt mit dem Schlüsselwort PROCEDURE und endet mit der Bezeichnung der Prozedur oder einer Parameterliste. Der Prozedurrumpf beginnt mit dem Schlüsselwort IS und endet mit dem Schlüsselwort END gefolgt von der optionalen Prozedurbezeichnung. Der Prozedurrumpf besteht aus drei Teilen: einem deklarativen Teil, einem ausführbaren Teil und einem optionalen Teil zur Fehlerbehebung.

Syntax:

```
PROCEDURE name [ (parameter [, parameter,...]) ] IS
      [local declarations]
BEGIN
      executable statement
[EXCEPTION
      exception handlers]
END [name];
```

Der Befehl EXECUTE führt eine gespeicherte Prozedur mit den zugehörigen SQL-Anweisungen aus. An die gespeicherte Prozedur können Parameter übergeben werden.

5.2 Designer /2000

Oracle Designer ist ein CASE (Computer Aided Systems Engineering) Werkzeug von Oracle und erlaubt den Projektstart in verschiedenen Phasen. In welcher Phase man startet und welches Verfahren man einsetzt, hängt im wesentlichen von den Voraussetzungen ab, die dem Projekt zugrundeliegen. Für diese Diplomarbeit wurden der *Entity-Relationship-Diagrammer* (ER-Diagrammer), der *Database-Design-Transformer* und der *Design-Editor* verwendet. Diese Werkzeuge werden später näher erläutert.

In Abbildung 7: wird die Vorgehensweise für die Anwendung InnoCon im *Designer/2000* dargestellt:

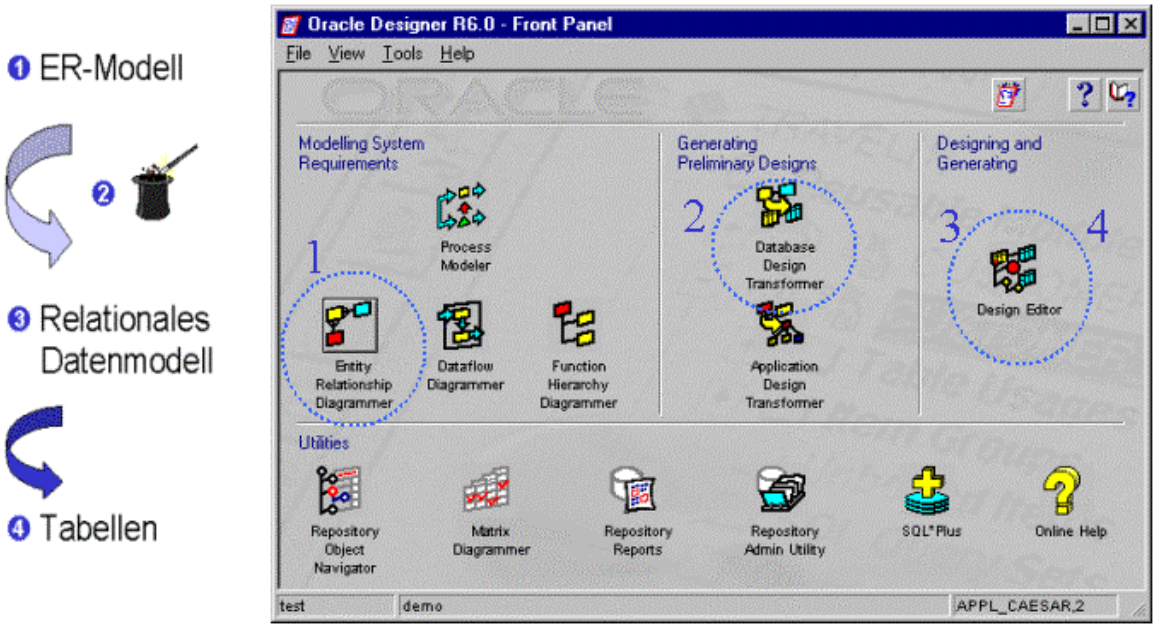


Abbildung 7: Vorgehensweise im *Designer/2000*

5.2.1 Entity-Relationship-Diagrammer

Für die Erstellung des ER-Modells wurde der *Entity-Relationship-Diagrammer* (ER-Diagrammer) von Oracle Designer/2000 benutzt (vgl. Abbildung 8: Entity-Relationship-Diagrammer). Der *ER-Diagrammer* dient zum graphischen Darstellen und Bearbeiten von Entitäten und Attributen. Die darin erstellten Datendiagramme entsprechen dem ER-Diagramm. Im *ER-Diagrammer* können auch Domains (Wertebereiche) festgelegt und zugewiesen werden.

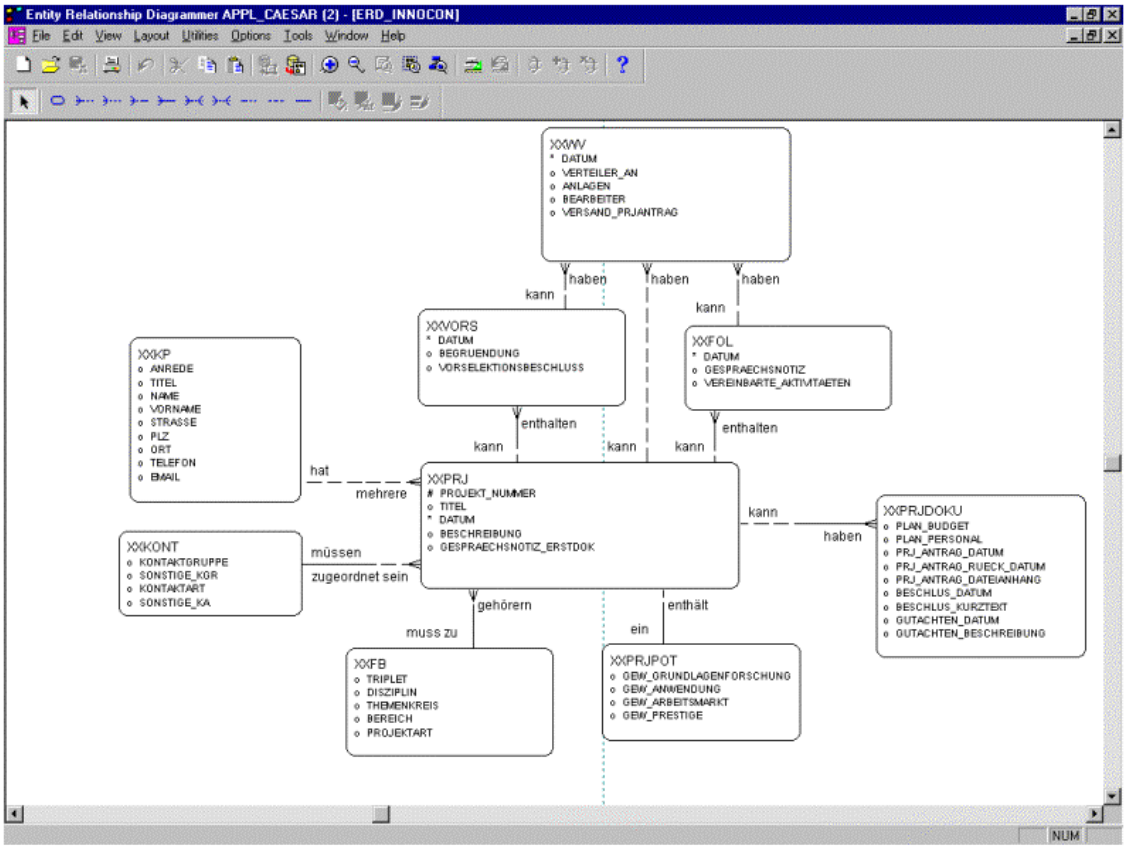


Abbildung 8: Entity-Relationship-Diagrammer

Nachdem in der Analysephase des Projektes die Entity-Relationship-Modellierung durchgeführt wurde, kann nun das entstandene Modell in ein relationales Datendesign überführt werden.

Die Trennung zwischen ER-Modell und Datenmodell sollte sehr exakt vorgenommen werden. Erst wenn ein stabiles ER-Modell entwickelt wurde, leitet man aus diesem das physische Datenmodell ab. Dieses kann später optimiert werden.

5.2.2 Database-Design-Transformer

Für das Überführen in ein relationales Datendesign wurde der *Database-Design-Transformer* (vgl. Abbildung 7: Vorgehensweise im Designer/2000, Punkt 2) eingesetzt.

Der *Database-Design-Transformer* führt folgende Ableitungen durch:

1. Entitäten werden zu Tabellen
2. Attribute werden zu Spalten
3. Relationen werden zu Fremdschlüsseln
4. Beidseitige Mehrfachbeziehungen (N:M) werden in Referenztabellen aufgelöst
5. Identifizierende Attribute werden zu Primärschlüsseln.
6. Logische Namen werden in physikalisch gültige Namen umgesetzt.
7. Entitäten ohne identifizierende Attribute können mit *Surrogatschlüssel* versehen werden. Hierbei handelt es sich um einen künstlich erzeugten Schlüssel, der allein zur Identifizierung der einzelnen Exemplare einer Entität dient. Dieser Schlüssel wird normalerweise durch eine einfache laufende Nummer gebildet. Ein Surrogatschlüssel ist immer auch ein Primärschlüssel.

Wichtig ist, dass nach der Transformation das ER-Modell keinen großen Änderungen mehr unterliegen darf, denn ein einmal abgeleitetes Datendesign lässt sich nicht immer automatisch an ein geändertes ER-Modell anpassen.

5.2.3 Design-Editor

Im *Design-Editor* erfolgt der logische Entwurf. Das logische Datendesign kann von Grund auf im *Design-Editor* neu entwickelt oder aus einem ER-Modell abgeleitet werden (vgl. Abbildung 9: Design Editor).

Im rechten Fenster des *Desing-Editors* ist das Server Modell Diagramm der Anwendnung InnoCon dargestellt.

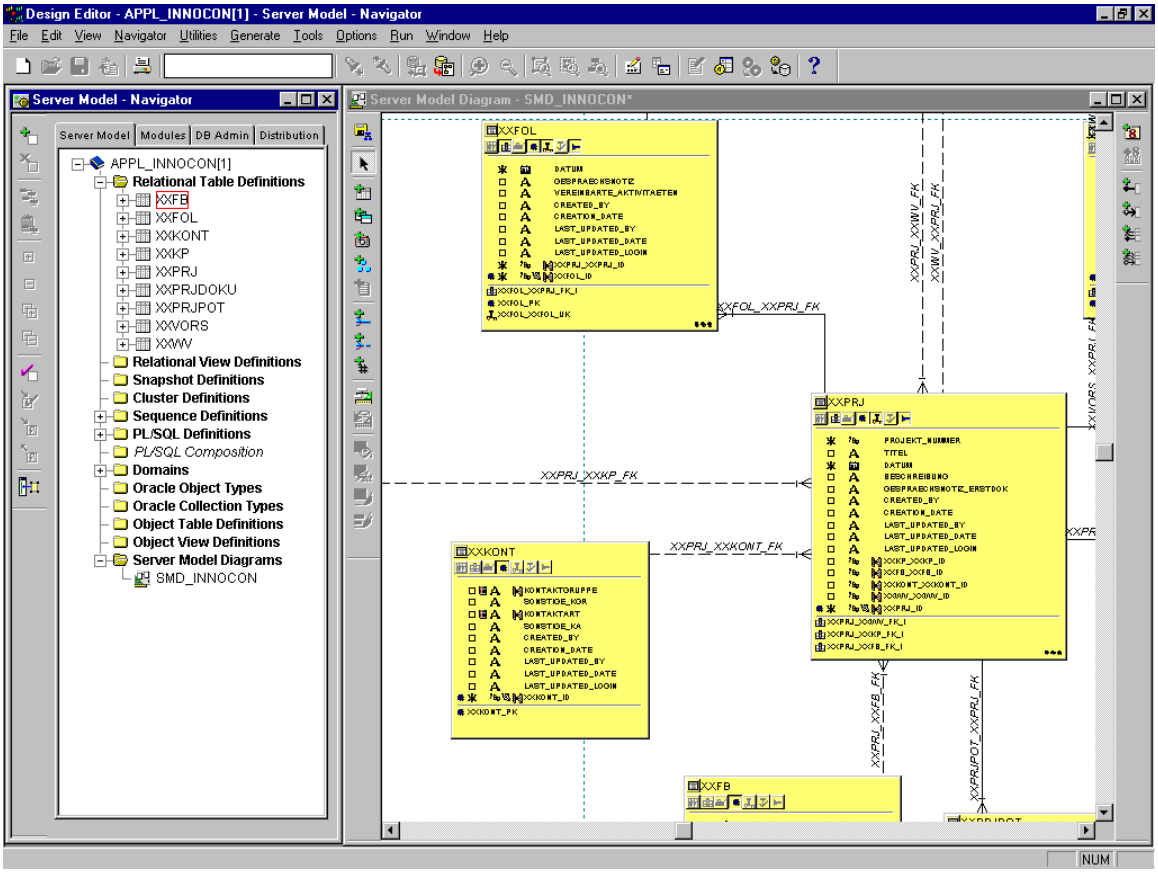


Abbildung 9: Design-Editor

Im logischen Datendesign erfolgt die Überprüfung auf Einhaltung der Normalformen.

5.2.3.1 Normalisierung der Relationen

In der folgenden Tabelle erfolgt die Spezifizierung der Datenobjekte (Tabellen).

Tabellenname	*Bedeutung*	*Beschreibung*
XXPRJ	*Projektidee*	*Tabelle für die Erfassung der Projektideen.*
XXKP	*Kontaktperson*	*Kontaktperson für diese Projektidee.*
XXPRJDOKU	*Projektdokumente*	*Projektdokumente, wie Gutachten oder Projektantrag.*
XXKONT	*Kontakte*	*Welche Kontakte führten zu der Projektidee?*
XXFB	*Forschungsbereich*	*Zu welchem Forschungsbereich gehört die Projektidee?*
XXVORS	*Vorselektion*	*Vorselektionsentscheidung über die Projektidee.*
XXFOL	*Folgedokumente*	*Folgedokumente zur Spezifizierung der Projektidee.*
XXWV	*Wiedervorlage*	*Die Projektidee, Vorselektionsentscheidung sowie die Folgedokumente können mit Hilfe der Wiedervorlage an anderen Mitarbeiter weitergeleitet werden.*

Tabelle 2: Tabellenbeschreibung InnoCon

Auf die Beschreibung der einzelnen Attribute wird im Kapitel 7 (vgl. Tabelle 5: Attributbeschreibung der Anwendung InnoCon) näher eingegangen.

Funktionale Abhängigkeiten

Primärschlüssel	→	**Nicht–Schlüssel–Attribute**
(Projekt_Nummer)	→	(Titel, Datum, Beschreibung, Gew_Grundlagenforschung, Gew_Anwendung, Gew_Arbeitsmarkt, Gew_Prestige)
(XXKP_ID)	→	(Anrede, Titel, Vorname, Name, Straße, PLZ, Ort, Telefon, Email)
(XXPRJDOKU_ID)	→	(Plan_Budget, Plan_Personal, Prj_Antrag_Datum, Prj_Antrag_Rueck_Datum, Prj_Antrag_Dateinanahang, Beschluss_Datum, Beschlus_Kurztext, Gutachten_Datum, Gutachten_Beschreibung)
(XXKONT_ID)	→	(Kontaktgruppe, Sonstige_KGR, Kontaktart, Sonstige_KA)
(XXFB_ID)	→	(Triplet, Disziplin, Themenkreis, Bereich, Projektart)
(XXVORS_ID)	→	(Datum, Begruendung, Vorselektionsbeschluss)
(XXFOL_ID)	→	(Datum, Gespraechsnotiz, Vereinbarte_Aktivitaeten)
(XXWV_ID)	→	(Datum, Verteiler_an, Anlagen, Bearbeiter, Versand_Projektantrag)

Alle aufgeführten Relationen sind in der **3. Normalform,** da sie sich in der 1NF und in der 2NF (funktional vom Primärschlüsselattribut abhängig) befinden und da kein Nicht–Schlüssel-Attribut transitiv abhängig von einem Schlüsselattribut ist.

5.2.4 Server-Generator

Der *Server-Generator* erzeugt aus den Definitionen des logischen Datenmodells Befehlsdateien mit Anweisungen zum Aufbau der Datenbank und der Datenstrukturen. Die Befehle werden in SQL formuliert. Der *Server-Generator* erzeugt ausschließlich DDL-Befehle zum Anlegen der Objekte, die im *Design-Editor* angelegt worden sind.
Da für die Anwendung InnoCon nur die DDL-Dateien generiert wurden, erfolgt das Implementieren der Objekte durch das Einspielen der SQL-Dateien in die Applikation mit dem Programm SQL*Plus.

5.3 Developer/2000

Das Konzept einer Datenbankanwendung kann bis zur Erstellung der Masken und der Reports vollständig mit dem *Oracle Designer/2000* realisiert werden. Die Masken sowie die Berichte (Reports) müssen jedoch mit dem Entwicklungswerkzeug *Oracle Developer/2000* erstellt werden.

Mit dem *Developer/2000* können Systeme entwickelt werden, die auf der Technologie einer grafischen Benutzeroberfläche (GUI), einer Datenbank, einer Client-Server- und einer Web-Umgebung basieren.

Oracle Developer/2000 besteht hauptsächlich in erster Linie aus den vier Komponenten: *Oracle-Forms, Oracle-Reports, Oracle-Graphics und Procedure-Builder.*

Der folgende Abschnitt behandelt die Entwicklungswerkzeuge *Forms-Designer* und *Reports-Builder* des Developer/2000.

5.3.1 Forms-Designer

Mit dem *Forms-Designer* können interaktive Anwendungen wie Masken (vgl. Abbildung 10: Forms-Designer), Menüs und Bibliotheken entwickelt werden. Bibliotheken bestehen aus PL/SQL-Prozeduren, Funktionen oder Paketen.

Die meisten Forms-Anwendungen bestehen in erster Linie aus Formularen mit einem oder mehreren Menüs und mehreren Bibliotheken. Die entworfenen Module werden standardmäßig in einer Datei gespeichert. Es besteht auch die Möglichkeit, die Module in der Datenbank zu speichern, dazu müsste jedoch das benötigte Datenbank-Schema für den *Forms-Designer* in die Datenbank geladen werden.

Zum Entwurf von Formularen wird der *Forms-Designer* eingesetzt. Formulare werden im *Forms-Designer* als binäre .fmb-Dateien gespeichert, diese können auf verschiedene andere Plattformen portiert werden. *Forms-Generator* wird aus dem *Forms-Designer* aufgerufen, um aus einer .fmb-Dateien eine .fmx-Datei zu erzeugen, die von *Forms-Runtime* ausgeführt wird. Die folgenden Abbildung (vgl. Abbildung 10: Forms-Designer) zeigt das Formular für die Erfassung der Projektidee (XX_PRJ):

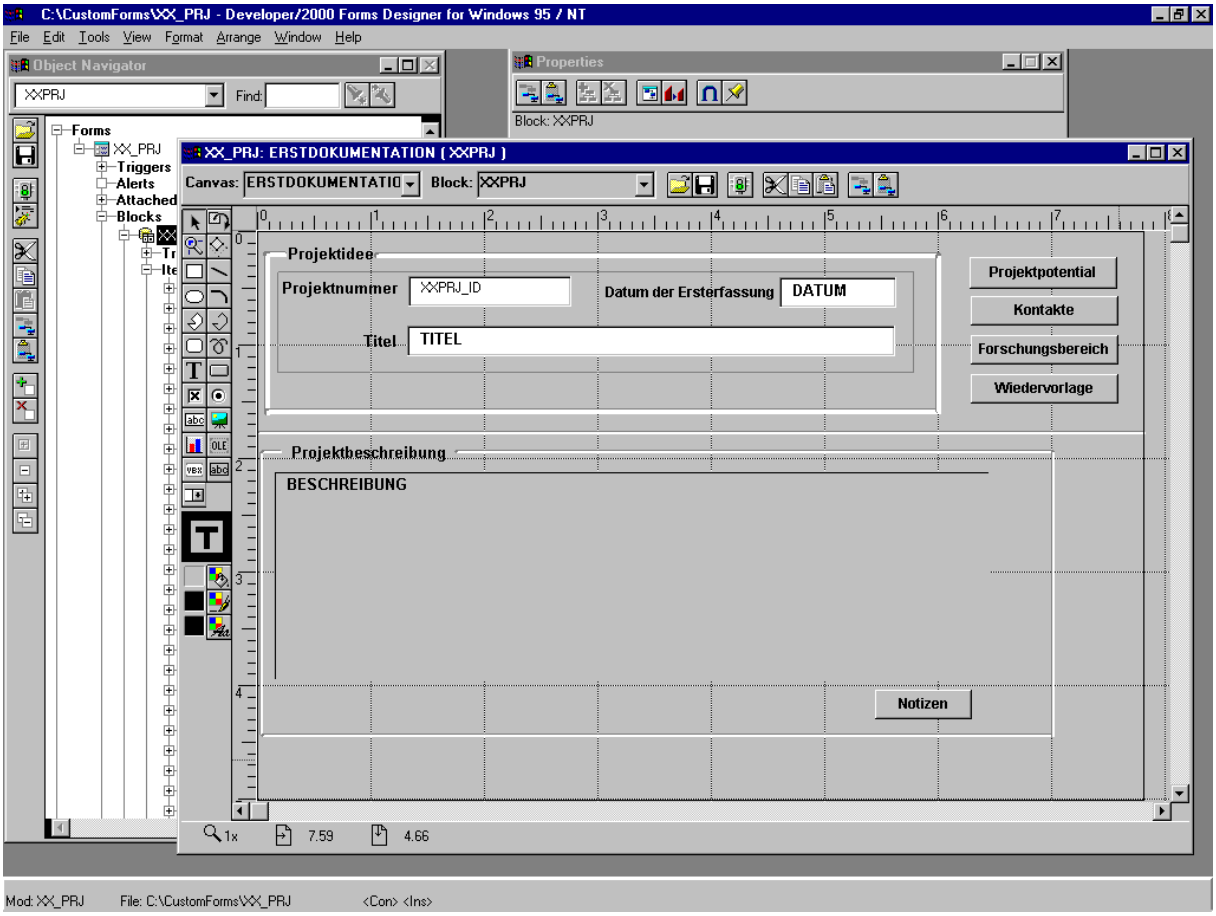


Abbildung 10: Forms-Designer

Ein Formular besteht aus folgenden Objekten:

1. Blocks (Blöcke):

   Jedes Formular enthält mindestens einen Block. Blöcke können entweder Basistabellenblöcke (ein Basisblock hängt mit einer Tabelle oder Datensicht zusammen) oder Steuerblöcke sein (ein Block hängt nicht mit einer Tabelle oder Datensicht zusammen, sondern enthält Steuerobjekte wie Schaltflächen).

   Blöcke können auch Master- und Detail-Blöcke sein, die ein Master-Detail-Formular erzeugen, d.h. im Master-Block wird ein Datensatz und im Detail-Block werden die zugehörigen Detail-Datensätze angezeigt.

2. Canvas (Ansichten):

   Ein Canvas ist ein Objekt im Hintergrund, auf dem Blockelemente platziert sind. Jedes Canvas besitzt eine Ansicht, in der festgelegt ist, wie das Canvas erscheint. Mit dem Layout-Editor lässt sich diese Ansicht im *Forms-Designer* bearbeiten. Abbildung 10: zeigt das Canvas „Erstdokumentation“.

3. Fenster:

   Ein Fenster ist ein leerer Rahmen, der über der Canvas-Ansicht liegt. Jedes Formular hat mindestens ein Fenster, aber es kann auch mehrere Fenster enthalten.

4. Trigger:

   Ein Trigger besteht aus einer oder mehreren PL/SQL-Anweisungen, die beim Auftreten der triggerauslösenden Aktion ausgeführt werden. Es können Trigger definiert werden, die beim Aktivieren eines Formulars ausgelöst werden oder beim Drücken einer Schaltfläche. Möchte man nach dem Betätigen der Schaltfläche „Neue Projektidee“ in der Maske „Projektideen“ eine Nummer vergeben, wird ein WHEN-BUTTON-PRESSED Trigger definiert, der folgende Syntax enthält:

Syntax:

```
CLEAR_FORM;                          -- leert den Inhalt der Maske
XXPRJ_CREATE_XXPRJ_ID;               -- Prozeduraufruf
:XXPRJ.CREATION_DATE := SYSDATE;     -- Datum einfügen
:XXPRJ.DATUM := SYSDATE;             -- Datum einfügen
```

Der Trigger ruft eine vordefinierte Prozedur auf, in der die Projektidee mit Hilfe der Sequenz erstellt wird:

Syntax:

```
PROCEDURE XXPRJ_CREATE_XXPRJ_ID IS
BEGIN
  SELECT INNOCON.XXPRJ_SEQ.NEXTVAL INTO :XXPRJ_ID
    FROM sys.DUAL;
END XXPRJ_CREATE_XXPRJ_ID;
```

5. Datensatzgruppen:

Datensatzgruppen (RECORD GROUPS) sind Gruppen von Werten, die wie folgt definiert werden können.

a) Dynamisch, durch eine SELECT-Anweisung, die zur Laufzeit ausgeführt wird.

b) Statisch, durch eine Gruppe statischer Werte

c) Programmgesteuert, durch ein PL/SQL-Unterprogramm, das zur Laufzeit eine SELECT-Anweisung oder eine Gruppe von Werten aufbaut.

6. Wertelisten:

Wertelisten (LOVs) sind auf Datensatzgruppen basierende Objekte, die zur Anzeige einer Liste mit möglichen Werten eingesetzt werden.

Abfragen erfolgen mit Erstellen einer Record-Gruppe. Dies ist eine SELECT Anweisung auf die benutzte Datenbank-Tabelle. Aus der Record Gruppe wird ein LOV (List of Values) erstellt. Dieser LOV wird einer Spalte in der Maske zugeordnet, dadurch kann die Abfrage der bereits eingegebenen Daten erfolgen. LOVs können auch als Rückgabewert oder als Suchfunktion eingesetzt werden.

5.3.2 Oracle-Reports

Oracle-Reports besitzt wie auch *Oracle-Forms* einen Objekt-Navigator, der zur Erstellung und Anpassung von Berichtselementen verwendet wird. Im Objekt Navigator werden Objekte in hierarchischer Anordnung dargestellt. Ein Oracle-Bericht besteht aus folgenden Elementen: Datenmodel, Layout, Parametermaske, Berichts-Triggern, Programmeinheiten und Verbundene Bibliotheken.

Oracle-Reports ist ein Werkzeug für den Entwurf von einfachen oder komplexen Berichten, die vor dem Ausdruck auf dem Bildschirm angezeigt werden können

Es gibt folgende Möglichkeiten, einen Report zu erstellen:

Auf einer einzigen Tabelle basierender Bericht:
Dieser Berichtstyp beschränkt sich auf die Verwendung nur einer einzigen Tabelle und gibt das Ergebnis der Berichtausführung in tabellarischer Form aus.

Master-/Detail-Bericht:
Dieser Berichtstyp ähnelt dem Master-/Detail-Formular aus *Oracle-Forms*. Auch hier gibt es eine Master-Abfrage und eine Detail-Abfrage.

Elemente eines Oracle-Berichts

<u>Datenmodell:</u>
Im Datenmodell erfolgt die Beschreibung der Abfragen, die Datensätze aus der Datenbank abrufen. Eine Abfrage kann aus jeder gültigen SELECT-Anweisung bestehen.

Das Datenmodel beschreibt die Abfragen, die Datensätze aus der Datenbank abrufen. Ein Datenmodell setzt sich aus einer oder mehreren Abfragen zusammen. Abfragen bestehen aus SELECT-Statements, die Daten aus verschiedenen Tabellen zusammenstellen können. Um die Reihenfolge anzugeben, wird in diesen SELECT-Anweisung die ORDER BY-Klausel verwendet.

Layout:

Berichte können bis zu vier Layoutkomponenten besitzen (Header, Body, Trailer, Margin), die wiederum aus vielen verschiedenen Elementen bestehen können wie Textfelder, Rahmen, Linien und Graphiken.

Der Header eines Berichtes wird zu Anfang eines Berichtes gedruckt. Im Header eines Berichtes könnten Informationen des Inhaltes über den Bericht dargestellt werden.

Oracle-Reports beinhaltet zwei relevante Werkzeuge: *Reports-Designer* und *Reports-Runtime*. Mit *Reports-Designer* können drei Modultypen erstellt und angepasst werden: Berichte, externe Abfragen und Bibliotheken. Genau wie Formulare des *Oracle-Designers* können auch Berichte entweder in der Datenbank oder im Verzeichnissystem gespeichert werden.

Reports werden als RDF-Dateien gespeichert. RDF steht für Reports Definition File. Dieses binäre Dateiformat ist in verschiedene Betriebssysteme portierbar. Der *Reports-Designer* generiert aus der RDF-Datei eine binäre REP-Datei, die zur tatsächlichen Ausführung eines Berichtes von *Reports-Runtime* verwendet wird.

5.3.2.1 Benutzerparameter für Reports

Mit einem Parameterformular ist es möglich, zur Laufzeit Benutzereingaben auszuwerten, um somit beispielsweise die Berichtsausgabe einzuschränken.
Mit Parametern können dynamische Berichtsdokumente entwickelt werden, um variable Berichtsausgaben zu produzieren. Die Berichtsausgaben sind von den Parametern abhängig, die während der Laufzeit eingegeben werden.
Ein benutzerdefinierter Parameter ist ein vom Entwickler des Reports erstelltes Datenmodell-Objekt, das einen Wert enthält, den der Benutzer zur Laufzeit ändern kann.
Parameter können an beliebigen Stellen in einer Abfrage eingesetzt werden, z.B.: Werte in der WHERE-Klausel einer SELECT-Anweisung einschränken, oder eine einzelne Spalte bzw. einen einzelnen Ausdruck in der SELECT-Liste ersetzten.
Parameter können auf zwei verschiedene Arten in einer Abfrage referenziert werden. Mit einer Bind-Referenz oder mit einer lexikalischen Referenz.

Eine **Bind-Referenz** ersetzt einen einzelnen Wert oder Ausdruck. Um eine Bind-Referenz in einer Abfrage zu erstellen, stellt man dem Parameternamen einen Doppelpunkt (:) voran.
Eine Bind-Referenz wird an beliebigen Stellen in einer Abfrage verwendet, wo ein einzelner Wert vorkommen kann, wie z.B. eine Zeichenkette, eine Zahl oder ein Datum.
Syntax:

```
SELECT XXPRJ_ID, TITEL
FROM innocon.XXPRJ
WHERE ID = :Projektnummer
```

Wenn man bei dieser Anweisung während der Laufzeit eine spezifische Projektnummer eingibt, verwendet die WHERE-Klausel den Wert der eingegebenen Projektnummer, um die abgerufenen Daten einzuschränken und die Projektidee mit der ausgewählten Nummer aufzurufen.

Eine **lexikalische Referenz** ersetzt einen beliebigen Teil einer SELECT-Anweisung, wie z.B. Spaltennamen, die Klauseln FROM, WHERE, ORDER BY. Um lexikalische Referenzen in einer Abfrage zu erstellen, stellt man dem Parameternamen ein kaufmännisches Und (&) voran.
Falls das Parameterobjekt nicht existiert, erstellt es Report Builder automatisch und zeigt eine Meldung an. In diesem Fall ist der Standarddatentyp eine Zeichenkette (CHARACTER) und keine Zahl (NUMBER).

5.4 FTP

Das File Transfer Protocol (FTP) dient zum Transport von Dateien via Internet oder Intranet.
Es existieren viele kostenlose FTP-Programme. Der Verfasser der vorliegenden Diplomarbeit hat sich für das Freeware FTP-Programm Leech-FTP entschieden. Mit Hilfe dieses FTP-Programms konnten die entwickelten Anwendungen, wie Forms und Reports, aus der Windows Umgebung auf den *Oracle Applications* Server transferiert werden.

6. Oracle Applications

Der Begriff *Oracle Applications* (Oracle Anwendungen) wird synonym zu *Oracle Financials* benutzt. Jedoch sollte hier eine Trennung der Begriffe erfolgen, da *Oracle Applications* nicht nur die Financials Module, sondern auch die Produktionsmodule (Manufacturing-Module) wie Einkauf (Purchase Order) und Lager (Oracle Inventory) umfasst.
Nach Angaben von Oracle ist die neue Bezeichnung für Oracle Applications offiziell Oracle eBusiness Suite. Dazu gehört auch Customer Relationship Management (CRM), welches aus 90 Modulen besteht.[41]

Oracle Financials umfasst verschiedene Module für betriebswirtschaftliche Anforderungen. Die Projektabrechnung (PA) wird ebenfalls unter *Oracle Financials* subsumiert, wenn es um den Bereich Projektfaktura geht. Auf die einzelnen Module wird im folgenden noch eingegangen.

6.1.1 Starten von Oracle Applications

Um *Oracle Applications* in der Stiftung caesar zu starten, ruft man auf der Client-Seite den Netscape Navigator mit der folgenden URL[42] auf:

http://erp0.caesar.de:8002/OA_HTML/jinitDEV.htm

Jede Seite im Web wird durch eine eindeutige **URL-Adresse** identifiziert, welche aus drei Komponenten zusammengesetzt ist.
Der Beginn der URL ist die Angabe des Protokolltyps für die Übertragung der Daten. Das Hypertext Transfer Protokoll ist das Standardprotokoll im Web und wird mit http abgekürzt.

[41] Vgl: http://www.oracle.de
[42] Uniform Resource Locator (Adresse eines Objekts im Internet)

<erp0.casar.de:8002> bildet den **Domainnamen** des Webservers und der TCP/IP-Port-Nummer *<8002>*. Mit der Portnummer wird die Datenbank DEV aufgerufen, auf der der Web Listener Anforderungen akzeptiert, während *<OA_HTML>* den virtuellen Anwendungspfad von *Oracle Applications* darstellt in der die Datei *<jinitDEV.htm>* gespeichert ist.
Angaben, die am Ende der URL stehen, geben den genauen Pfad und den Namen des Dokumentes auf dem Webserver an.
<jinitDEV.htm> ist eine HTML-Datei in der die Parameter für den Aufruf *von Oracle Applications* gespeichert sind.

Im folgenden wird der Begriff ERP-Sever verwendet. Damit ist der Server auf dem die Datenbank sowie *Oracle Applications* installiert sind, gemeint.

6.1.2 Oracle Web Application Server (OWS)

Die URL-Adresse wird im *Web Application Server*, dem Web Server von Oracle, vorgegeben. Diese Vorgaben für den Zugriff auf eine Datenbank mittels Intranet oder Internet werden nach der Installation des *Web Application Servers* vorgenommen. Es wird eine Database-Access-Descriptor (DAD) erstellt und konfiguriert, über den der Zugriff auf verschiedene Datenbanken ermöglicht wird.

In der Regel werden verschiedene Instanzen installiert, um die Entwicklungs-umgebung (DEV) von der Produktionsumgebung (PROD) zu trennen.
Der *Web Application Server* startet den Web-Listener für den Zugriff auf die Objekte einer Datenbank. Dafür wird in der Konfiguration der DAD der Pfad der Datenbank angegeben. Es handelt sich um einen virtuellen Pfad und nicht um die Angabe eines physischen Pfades. Die Zuordnung zwischen virtuellem und physikalischem Pfad ist auf dem ERP-Server hinterlegt.
Nachdem der Client die URL-Adresse von *Oracle Applications* im Netscape Navigator aufruft, startet der *Oracle JInitiator*.

6.1.3 Oracle JInitiator

Da *Oracle Applications* in der Version 11.03 in einer Java–Umgebung läuft, muss auf der Client Seite der *Oracle JInitiator 1.1.7.18* für Netscape Navigator installiert werden. Der *JInitiator* wird als ein Plugin im Netscape Navigator oder als ActiveX-Komponente im Microsoft Internet Explorer installiert und wird benötigt um Java-Applets auf dem Client starten zu können.
Falls der *JInitiator* auf dem Client nicht installiert ist, wird der Client aufgefordert, dieses Plugin vom Host auf den eigenen Rechner zu kopieren und zu installieren. Nach der Installation hat der Client Zugriff auf Oracle Applications.

6.1.4 Aufbau von Oracle Applications

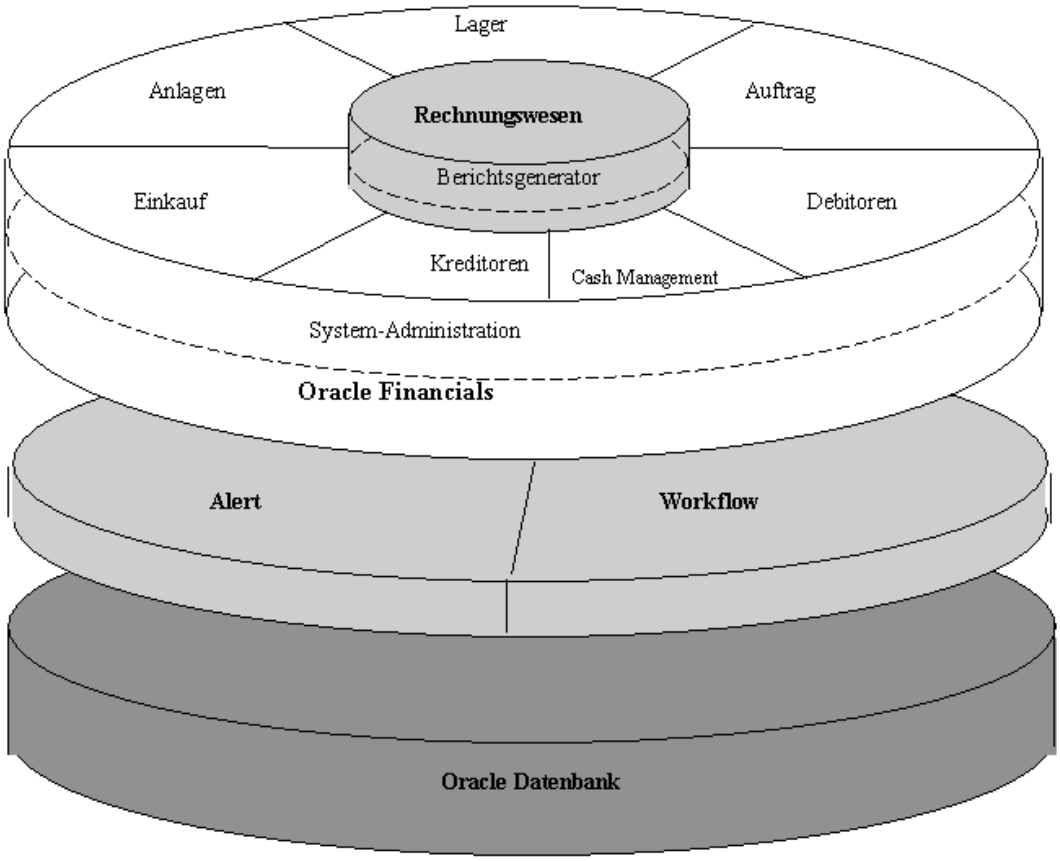


Abbildung 11: Die wichtigsten Module der Oracle-Anwendungen

Oracle Financials (Finanzanwendungen)

Zum Bereich der Finanzanwendungen gehören (vgl. Abbildung 11: Die wichtigsten Module der Oracle-Anwendungen) die Module Kreditoren, Debitoren, Anlagen, Rechnungswesen und Cash Management. Mit diesen Finanzmodulen können die zum Produktionsbereich gehörenden Module kombiniert werden. Die Produktionsmodule umfassen Oracle Einkauf, Oracle Lager und Oracle Auftrag.

Systemadministration

Das Modul Systemadministration verwaltet die Systemeinstellungen für die anderen Module: Hier werden Benutzer angelegt, Zuständigkeiten definiert oder Drucker gesteuert.

Die Oracle-Anwendungen nutzen eine Oracle-Datenbank (Version 8), um die eingegebenen Daten zu speichern. Zwischen der Oracle Datenbank und den Anwendungen liegt die Ebene der Alerts und Workflow, die bestimmte Prozessabläufe zwischen der Anwenderebene und der Datenbank steuert.

Oracle Workflow

Oracle Workflow ist ein integriertes E-Mail-System in *Applications*. Damit können Informationen beliebiger Art an Personen innerhalb des Unternehmens weitergeleitet werden. Workflows werden mit dem Workflow Builder definiert, einer graphischen Windows-Oberfläche, mit der die Benutzer den Geschäftsprozess, die Aktivitäten, Elemente, Nachrichten und Suchbegrifflisten sowie Rollen (Genehmigungshierarchie) entwerfen können. Dieser Workflow wird dann in den geschäftlichen Transaktionsprozess (Ablauf) integriert.

Alert

Oracle Alert ist ein System zur Erfassung von Ausnahmezuständen in der Datenbank. Mit Oracle Alert kann der Informationsfluss im Unternehmen über sogenannte Alerts kanalisiert werden. Damit werden Geschäftsdaten überwacht. Außerdem wird durch *Oracle Alert* sichergestellt, dass man nur die Informationen erhält, die man benötigt. Es können die vorkodierten Alerts modifiziert oder eigene erstellt werden.

Ein Alert-Vorgang kann beinhalten, dass eine E-Mail direkt an das im Unternehmen integrierte E-Mail-System gesendet wird, ein Hintergrundprogramm, ein Betriebssystemskript oder ein SQL-Skript gestartet wird. Es können zwei Arten von Alerts definiert werden:

1. Ereignis-Alerts

Ein Ereignis-Alert sendet eine Benachrichtigung in dem Moment, in dem ein bestimmtes Ereignis in der Datenbank auftritt. Dieses Ereignis könnte beispielsweise eine Einfügung oder Änderung in einer bestimmten Datenbanktabelle sein.

Durch die Erstellung von Ereignis-Alerts erhält man einen aktuellen Einblick in die Aktivitäten der Datenbank, da man stets über wichtige oder ungewöhnliche Ereignisse unterrichtetet wird.

2. Periodische Alerts

Im Gegensatz zum Ereignis-Alert überprüft ein periodischer Alert die Datenbank nach einem definierten Zeitplan. Periodische Alerts können beliebig oft durchgeführt werden. Mit einem periodischem Alert können Routinevorgänge, wie z.B. die Erbringung von aktuellen Nachweisen über die Leistung der Mitarbeiter oder die Erfassung potentieller Problempunkte der Datenbank automatisiert und in regelmäßigen Abständen wiederholt werden.

6.2 Oracle Financials - Ein Überblick

Oracle Financials ist eine Enterprise-Resource-Planning Software.
Enterprise-Resource-Planning (Unternehmensressourcenplanung) ist ein Begriff aus der Betriebswirtschaft, der sich auf betriebliche Planung, Buchführung und das Management von Unternehmensressourcen wie Lagerhaltung, Einkauf und Absatz bezieht.
ERP wird oft auch als Kürzel für ERP-Programme verwendet, Softwaresysteme, die diese betrieblichen Aufgaben unterstützen und den gesamten Planungs- und Steuerungsprozess eines Unternehmens über alle Wertschöpfungsstufen abbilden können.
Ein ERP-System besteht normalerweise aus einer oder mehreren relationalen Datenbanken sowie Anwendungsprogrammen, die für die einzelnen betrieblichen Aufgabenbereiche, wie Lagerverwaltung, Pflege der Kunden- und Personaldateien, Einkaufsplanung, Finanzbuchhaltung, Rechnungslegung und Produktionsplanung eingesetzt werden. Andere bekannte Anbieter von ERP-Systemen sind *SAP R/3* und *Baan*.

6.2.1 Modulstruktur

Oracle Financials besteht aus mehreren Funktionsgruppen, den sogenannten „Modulen“. So umfasst das Modul „Kreditoren“ folgende Funktionen: Lieferanten, Rechnungen, Zahlungen und bestimmte Auswertungen. Das Modul „Debitoren“ dagegen beinhaltet die Funktionen Kunden, Rechnungen und Zahlungseingänge. In der Abbildung 12: werden die wichtigsten Module der Oracle Anwendungen gezeigt.

6.2.2 Aufbau der Finanzanwendungen

Im Mittelpunkt der Oracle-Finanzanwendungen steht das Modul Rechnungswesen, auch Hauptbuch oder *General Ledger (GL)* genannt. In diesem Modul werden alle Transaktionen aufgezeichnet, die aus den Nebenbüchern stammen oder direkt im Rechnungswesen-Modul selber eingegeben werden (z.B. manuelle Buchungen).

Die Nebenbücher bestehen aus Oracle Kreditoren, Oracle Debitoren und Oracle Anlagen. Die Module arbeiten voneinander unabhängig, sind aber alle mit Oracle Rechnungswesen durch eine interne Verknüpfung, der sogenannten Schnittstelle, verbunden. Die Bankenbuchhaltung (*Cash Management*) besitzt interne Schnittstellen zu Oracle Kreditoren und Oracle Debitoren. Oracle Kreditoren wiederum ist über eine interne Schnittstelle mit der Anlagenbuchhaltung (Fixed Assets) verbunden (siehe Abbildung 12: Module der Oracle-Finanzanwendungen und ihre Verbindungen).

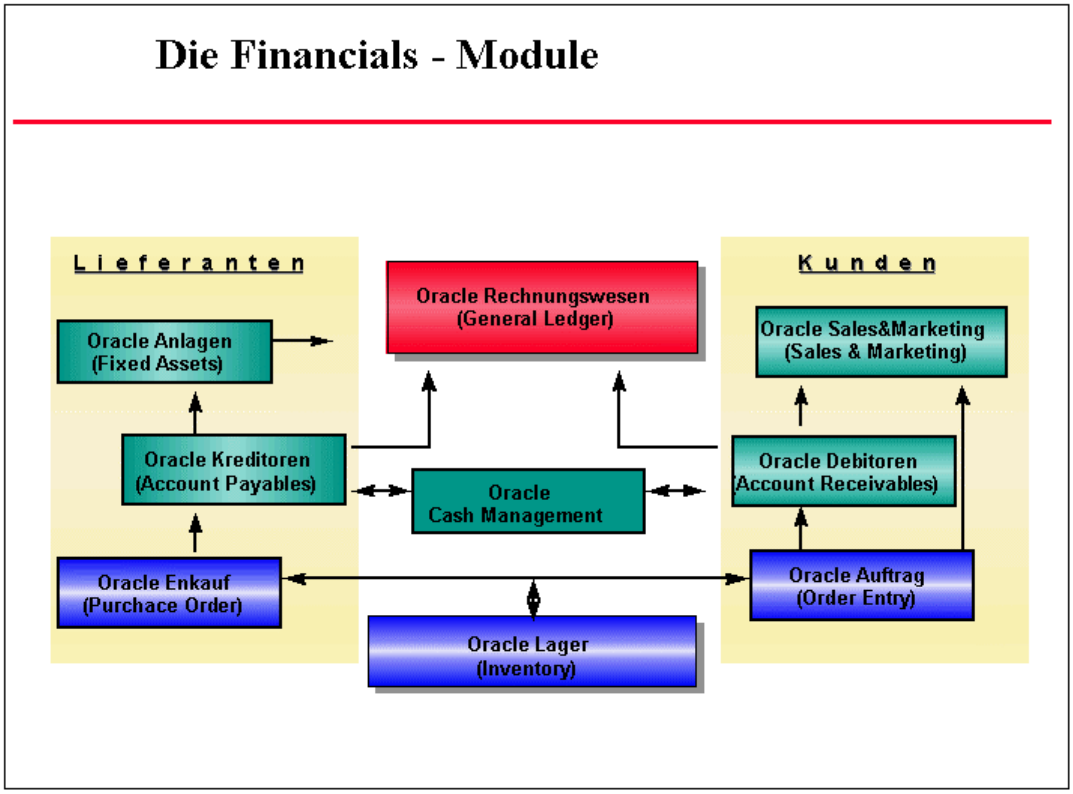


Abbildung 12: Module der Oracle-Finanzanwendungen und ihre Verbindungen

Oracle Kreditoren und Oracle Debitoren können mit der Materialwirtschaft verbunden werden, falls diese eingerichtet ist. In einer solchen Implementierung werden Materialbestellungen direkt im System erfasst und können in die Daten der Finanzbuchhaltung integriert werden.
Wie in Abbildung 13 dargestellt, ermöglicht das Modul Oracle Cash Management eine modulübergreifende Bearbeitung von Bankauszügen. Dieses Modul dient dazu, Bankauszüge mit Zahlungseingängen (Oracle Debitoren) und Zahlungsausgängen (Oracle Kreditoren) und mit sonstigen Zahlungen (Zinsen, Bankgebühren) abzustimmen.

Kontoflexfeld

Im Zentrum der Finanzanwendungen steht das Modul Rechnungswesen. Alle Transaktionen, die aus den Nebenbüchern stammen und Buchungen erzeugt haben, werden hier mittels des *Kontoflexfelds* aufgezeichnet. Unter Kontoflexfeld versteht man die Struktur, in der Konten, Kostenstellen und andere Informationen im Rechnungswesen aufgezeichnet werden. Das Kontoflexfeld teilt sich in mehrere Segmente auf. Ein ausgefülltes Kontoflexfeld bildet eine Kontenkombination. Ein Kontoflexfeld muss vollständig ausgefüllt sein, damit es über die interne Schnittstelle an Oracle Rechnungswesen übergeben und dort analysiert werden kann.

6.2.3 Oracle Financials - Die Navigation

Die Oracle Finanzanwendungen bieten eine graphische Benutzeroberfläche. Für die Eingaben wird in einigen Bereichen der Windows-Standard benutzt, einige Tastenkombinationen sind jedoch Oracle-spezifisch (beispielsweise das häufige Benutzen der Tab-Taste oder das Zeichen % als Platzhalter).

Die folgenden Abschnitte geben Hinweise für den Umgang mit den Oracle Anwendungen. Erklärt wird die Bedeutung von Zuständigkeiten, der Umgang mit dem Navigator sowie die Hintergrundprozesse.

6.2.3.1 Anmeldung

Um mit Oracle Financials arbeiten zu können, muss man sich im System anmelden:

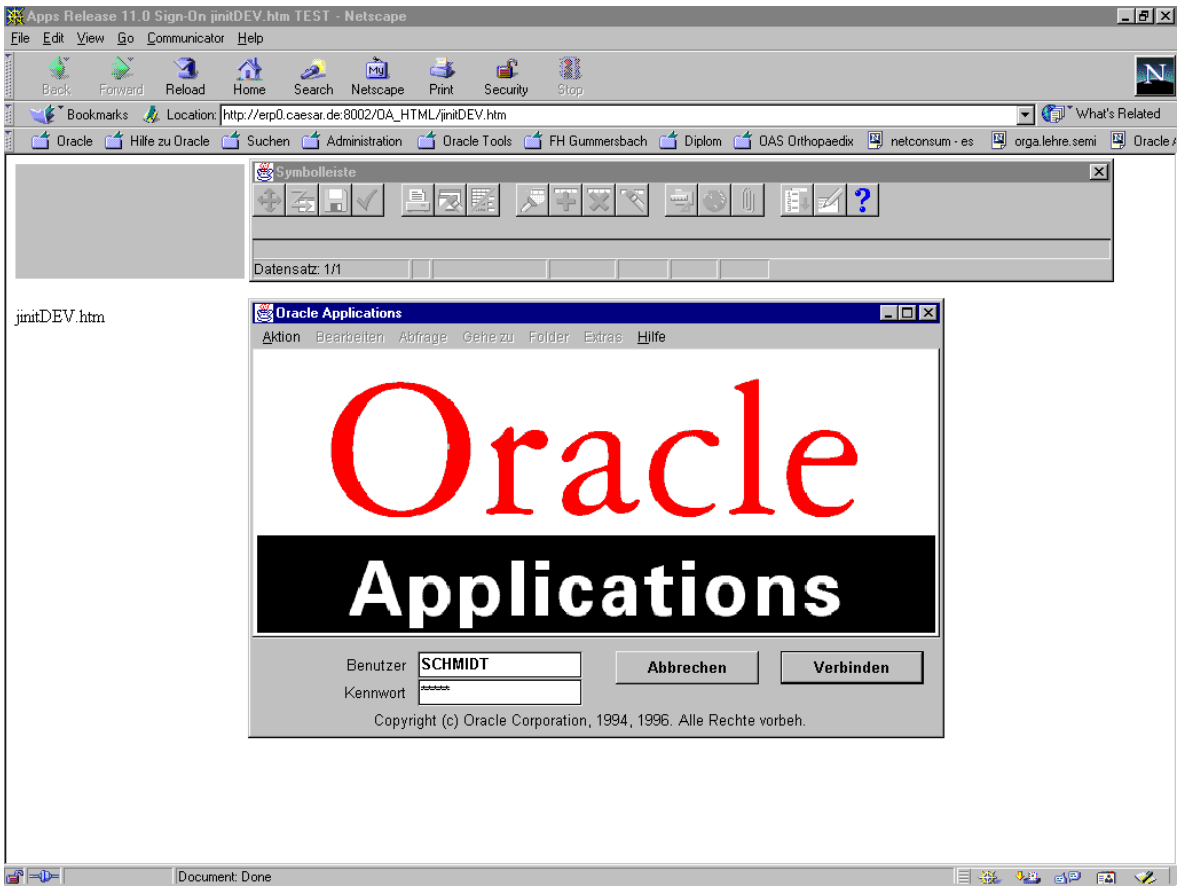


Abbildung 13: Die Anmeldemaske

6.2.3.2 Zuständigkeiten

Nach der Eingabe des Benutzernamens sowie des Passwortes erhält der Benutzer die ihm zugeordnete Zuständigkeit (vgl. Abbildung 14: Liste der Zuständigkeit). Unter Zuständigkeit ist der Zugriff auf verschiedene Module zu verstehen. Zuständigkeiten sind die eigentlichen Rollenverteilungen. Hier werden den einzelnen Aufgaben die entsprechenden Masken, Menübäume und Prozesse sowie Berichte zugeteilt.

Jede Zuständigkeit steht für bestimmte Funktionen. Öffnet man also eine Kreditoren-Zuständigkeit, kann man kreditorische Vorgänge durchführen. Durch das Öffnen einer Debitoren-Zuständigkeit dagegen bekommt man Zugriff auf debitorische Transaktionen. Für jeden Anwender werden vom Systemverwalter Zuständigkeiten vergeben.

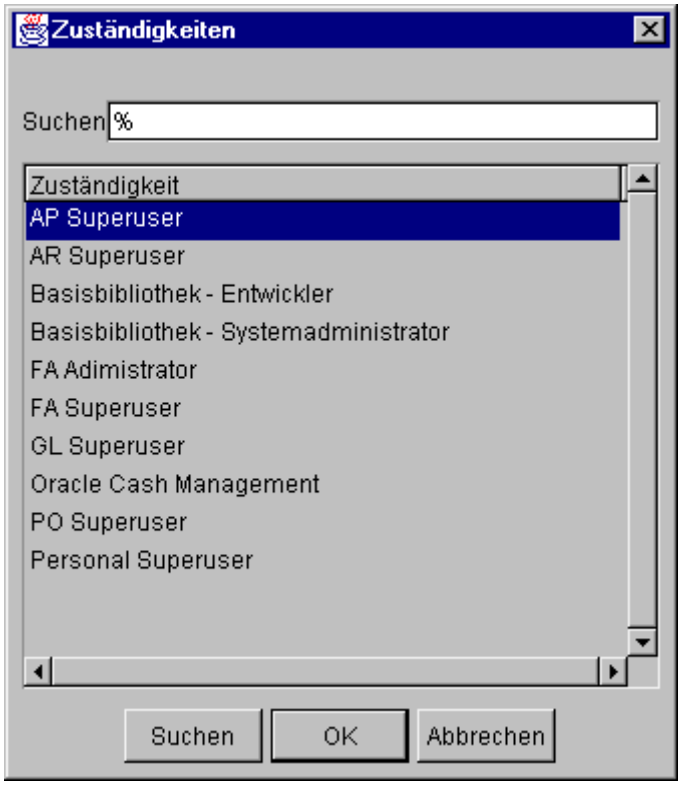


Abbildung 14: Liste der Zuständigkeiten

Falls einem Anwender vom Systemverwalter nur eine Zuständigkeit zugewiesen wurde, sieht dieser Benutzer keine Liste der Zuständigkeiten. In diesem Fall öffnet sich sofort der Navigator, der im nächsten Abschnitt beschrieben wird.

6.2.3.3 Der Navigator

Unmittelbar nachdem die Zuständigkeit ausgewählt wurde, öffnet sich der *Navigator* (vgl. Abbildung 15: Navigator). Der Navigator bietet die Möglichkeit, sich von einer Funktion zur nächsten zu bewegen.

Da jede Zuständigkeit andere Funktionen beinhaltet, hat der Navigator in jeder Zuständigkeit andere Inhalte. Diese werden in einem Menü dargestellt. In der folgenden Abbildung sieht man ein Beispiel für das Menü der Zuständigkeit Kreditoren (AP Superuser):

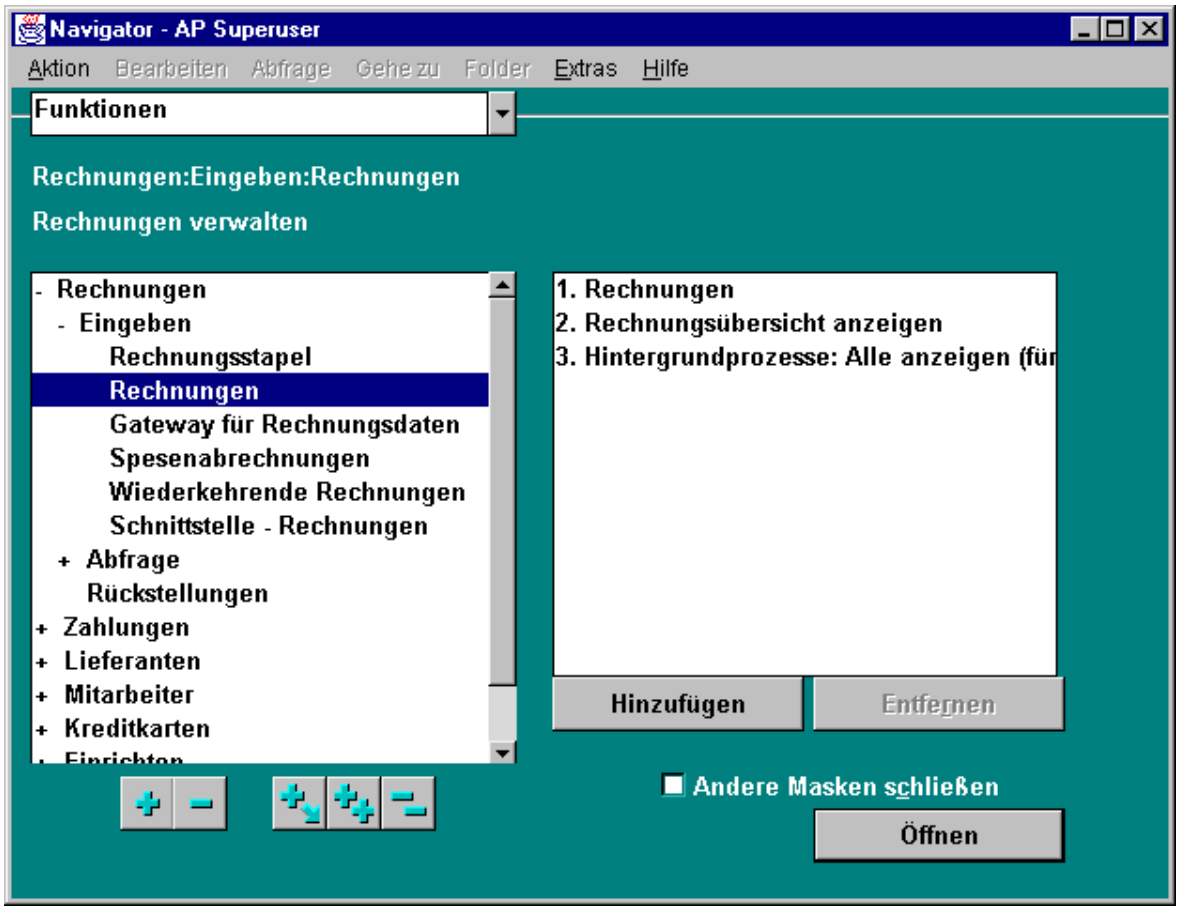


Abbildung 15: Navigator

Die Bedienung des Navigators ist, unabhängig von den jeweiligen Funktionen in jeder Zuständigkeit gleich.

Der Navigator bildet eine bestimmte Menüstruktur ab. Es gibt Ober- und Untermenüs. So umfasst beispielsweise die Funktion *"Rechnungen"* die Untermenüs *"Eingeben"* und *"Abfrage"*. Sobald man die unterste Ebene eines Menüs aufgerufen hat, öffnet sich eine Maske, die die ausgewählte Funktion anbietet.

Top-Ten-Liste

Oft müssen regelmäßig die gleichen Masken benutzt werden. Um dann die Auswahl eines Menüpunktes zu erleichtern, können bis zu zehn Menüpunkte in der *Top-Ten-Liste* hinterlegt werden. Diese bildet das rechte große Feld im Navigator.

6.2.3.4 Symbolleiste in Oracle Applications

In der Symbolleiste von Oracle Applications sind wichtige und oft benutzte Funktionen in Form von Symbolen hinterlegt. Diese werden im folgenden Erklärt.

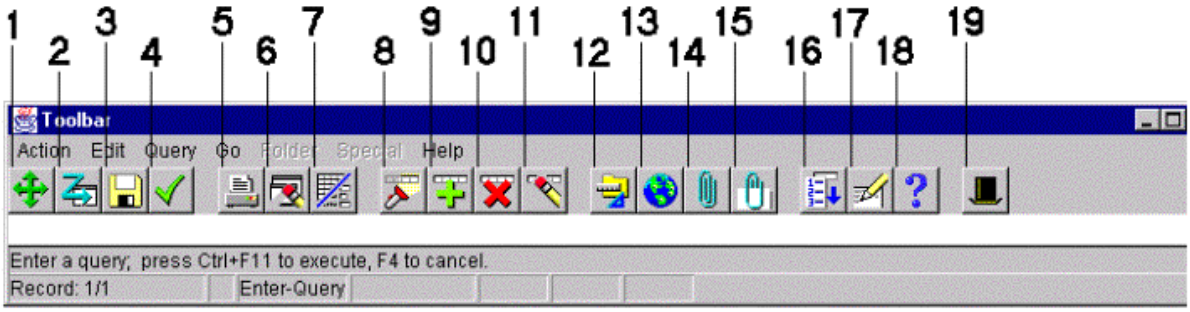


Abbildung 16: Oracle Applications - Symbolleiste

Nr.	Funktion	Menüleiste im Navigator
1	Ruft Navigator auf	Aktion > Navigieren
2	Ruft Zoom auf	Aktion > Zoom
3	Speichern	Aktion > Speichern
4	Speichern und Weiter	Aktion > Speichern und weiter
5	Drucken des geöffneten Fensters	Aktion > Drucken
6	Blendet alle Daten aus einer Maske aus, löscht Sie aber nicht	Bearbeiten > Maske ausblenden
7	Ruft Einzeldaten eines zusammengefaßten Datensatzes auf	Gehe zu > Übersicht/ Detail
8	„Taschenlampe“: Ruft spezielle Suchmasken auf	Abfrage > Suchen
9	„Grünes Kreuz“: Fügt einen zusätzlichen Datensatz ein	Bearbeiten > Neuer Datensatz
10	„Rotes Kreuz“: Löscht einen Datensatz.	Bearbeiten > Datensatz löschen
11	„Radiergummi“: Blendet einen Datensatz vom Bildschirm, aber nicht aus der Datenbank.	Bearbeiten > Datensatz

Nr.	Funktion	Menüleiste im Navigator
		ausblenden
12	Ruft Werkzeuge auf, um die Maske zu verändern und einen „Folder“ zu erstellen	Folder > Tools
13	Übersetzung in Fremdsprachen oder aus Fremdsprachen wird angezeigt	Bearbeiten > Fremdsprachen
14	Verknüpfung: Hinterlegte Dokumente können angesehen werden	Bearbeiten > Verknüpfungen
15	Verknüpfung: Dokumente können hinterlegt werden	Bearbeiten > Verknüpfungen
16	„Werteliste“: Eine Liste hinterlegter Werte kann geöffnet werden.	Bearbeiten > Werteliste
17	Editor: Feldinhalt wird in einem Texteditor angezeigt. Nützlich, wenn Texte länger als das Feld sind.	Bearbeiten > Feld bearbeiten
18	Hilfe: Ruft die online-Hilfe auf	Hilfe > Fenster
19	„Hütchen“: Wird nur im Navigator angezeigt. Ermöglicht das Wechseln in eine andere Zuständigkeit.	Extras > Zuständigkeit

Tabelle 2: Symbole in der Symbolleiste

Bedeutung von Hintergrundprozessen

Um Auswertungen (Berichte) aus den Oracle Finanzanwendungen zu erhalten oder auch Schnittstellenprogramme zu starten, wird die Funktion der *“Hintergrundprozesse“* genutzt. Die Aktionen zum Starten und Ansehen von Hintergrundprozessen sind in allen Zuständigkeiten identisch. Lediglich der Navigationspfad zum Einstieg unterscheidet sich in den Zuständigkeiten. Die genauen Navigationspfade werden vom Systemverwalter festgelegt. Hintergrundprozesse können einzeln oder im Verbund gestartet werden. Zudem kann der Startzeitpunkt von Hintergrundprozessen im voraus geplant werden.

Starten eines Hintergrundprozesses

Um Auswertungen zu starten, muss man über den Navigator die Maske zum Steuern von Hintergrundprozessen aufrufen. Es erscheint eine Abfrage, die nach der Art des zu startenden Prozesses fragt. Zur Auswahl stehen einzelne Prozessanforderungen oder -sets. Letztere sind Bündel von Prozessen, die gemeinsam gestartet werden. Die Erstellung eines solchen Sets ermöglicht es, mehrere Prozesse hintereinander ablaufen zu lassen, ohne sie einzeln starten zu müssen. Ein Prozess-Set kann mit Hilfe eines Assistenten oder auch manuell erstellt werden.

Nach der Bestätigung der Option *Einzelne Prozessanforderung* öffnet sich die Maske „*Hintergrundprozesse starten*“. Im Feld Prozessanforderung wird der zu startende Hintergrundprozess ausgewählt. (vgl. Abbildung 17: Starten eines Hintergrundprozesses).

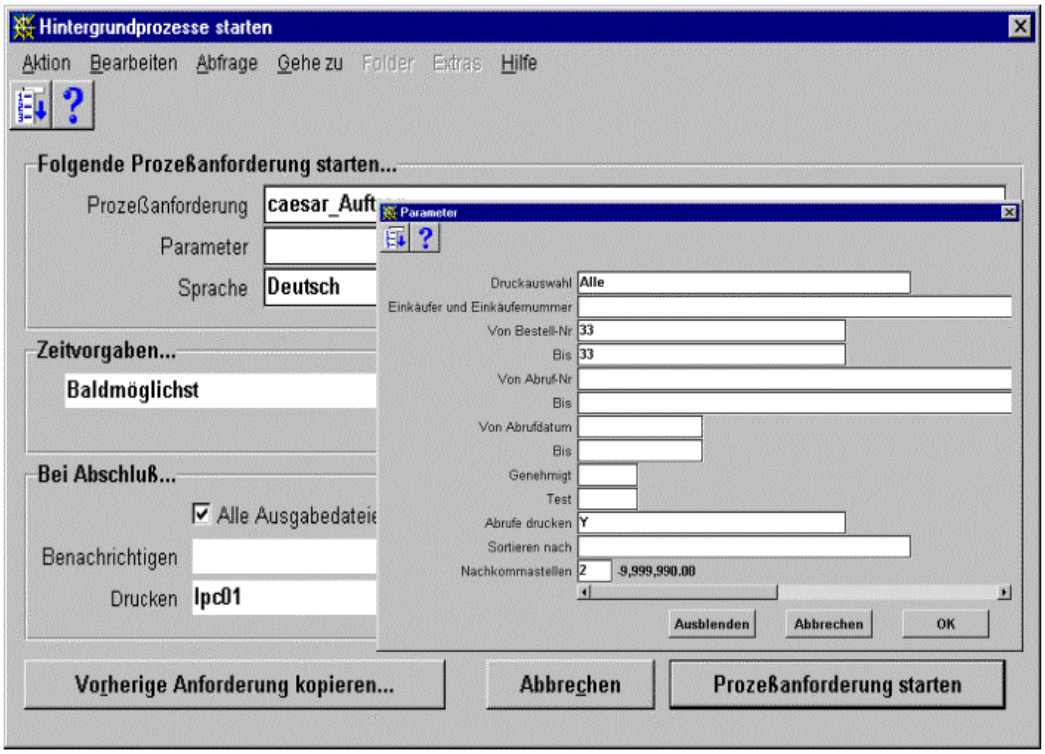


Abbildung 17: Starten eines Hintergrundprozesses

Nach der Auswahl öffnet sich automatisch das Fenster Parameter. Parameter sind je nach Report unterschiedlich und dienen zur Identifizierung des Berichtes, z.B. Ausdruck der Bestellung Nummer 33. Das obige Beispiel zeigt eine Parametermaske für den Report *caesar Auftrag* (vgl. Abbildung 17: Starten eines Hintergrundprozesses).

Als Hilfe zum Ausfüllen der Parameter steht in der linken oberen Ecke dieser Maske das Werteliste-Icon zur Verfügung. Hier kann man die Bestellnummer auswählen, für die der Report gestartet werden soll. Kann ein Parameter nicht mit Hilfe der Werteliste ausgefüllt werden, ist dieses Symbol grau.

Beim Starten jedes Hintergrundprozesses wird eine eindeutige Nummer vergeben, die sogenannte Prozessnummer. Diese dient zum Auffinden und Verwalten von Hintergrundprozessen. Falls man einen Hintergrundprozess aus einer Anwendungsmaske heraus startet, wird diese Nummer als Information in einer Dialogbox mitgeteilt, dann hat man die Möglichkeit, über die Maske „Hintergrundprozesse“ diesen Bericht anzusehen oder auszudrucken.

6.2.4 Sektor

Jedes Fenster in Oracle Financials besteht aus mindestens einem Sektor. Jeder Sektor enthält Daten, die sich auf eine bestimmte Unternehmenseinheit beziehen. In alle erforderlichen Felder eines Sektors müssen Daten eingegeben werden, damit man in den nächsten Sektor gelangen kann. Enthält ein Fenster nur einen Sektor, ist die Sektorbezeichnung identisch mit dem Fensternamen. Alle übrigen Sektoren eines Fensters haben eigene Namen. Jeder Sektor wird vom vorhergehenden Sektor durch eine horizontale Linie abgetrennt.

6.2.5 Datensatz

Ein Datensatz ist die Darstellung von gespeicherten Daten in den Feldern eines Sektors. Ein Datensatz wird manchmal auch als Datenzeile oder Transaktion bezeichnet, da ein Datensatz entweder einer Datenzeile in einer Datenbanktabelle oder einer logischen Transaktion entspricht.

Das Fenster "Hintergrundprozesse starten" (vgl. Abbildung 17: Starten eines Hintergrundprozesses) ist ein Beispiel für einen Sektor mit nur einem Datensatz. Wenn man sich den nächsten Datensatz anzeigen lassen möchte, wählt man im Menü "Gehe zu" den Befehl "Nächster Datensatz". Alle Feldwerte in diesem Sektor werden dann durch die Feldwerte des nächsten Datensatzes ausgetauscht.

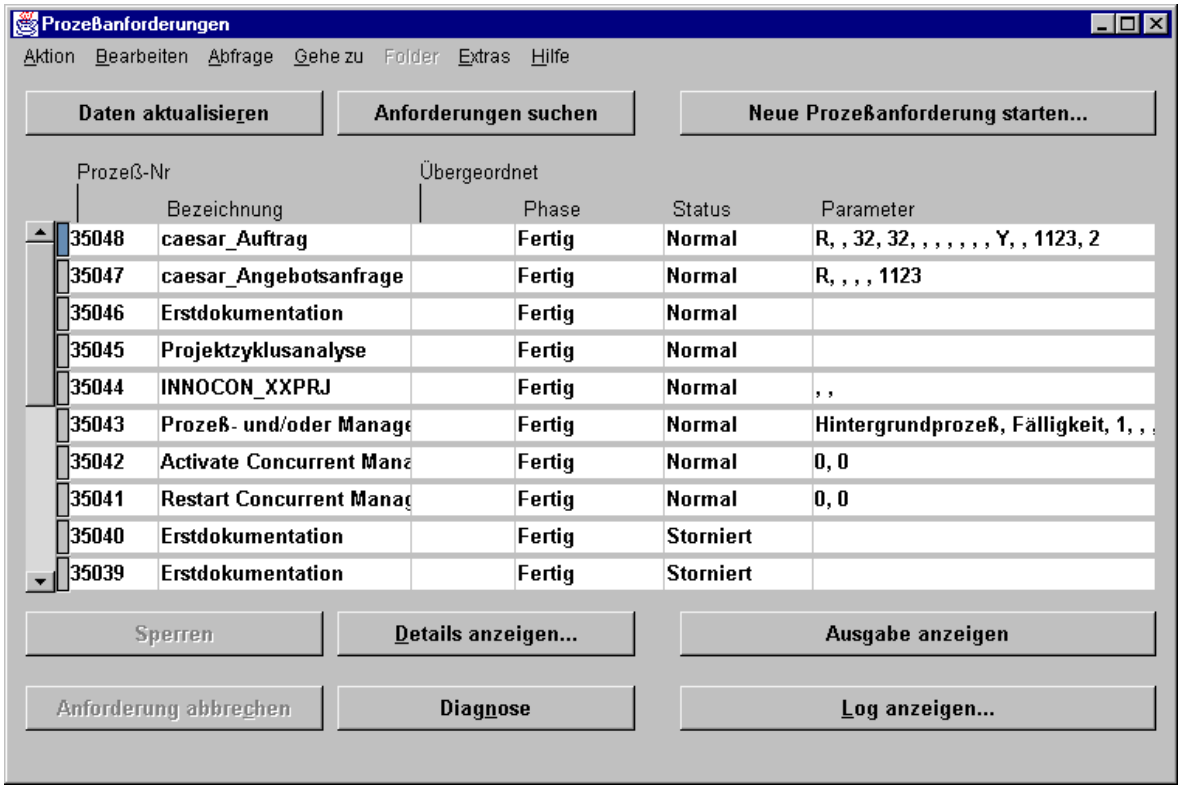


Abbildung 18: Prozeßanforderungen

Im Fenster "Prozeßanforderungen" (vgl. Abbildung 18: Prozeßanforderungen) hingegen werden bis zu zehn Datensätze gleichzeitig angezeigt. Jeder Datensatz stellt dabei einen Prozess dar, der vom Fenster "Hintergrundprozesse starten" gestartet wurde. Das Fenster "Prozesse" enthält nebeneinander angeordnete Felder, wobei unterhalb der Feldtitel die Daten aus zehn Datensätzen angezeigt werden können. Das Fenster "Prozesse" ist ein Beispiel für einen "Sektor mit mehreren Datensätzen".

6.2.6 Die Bedeutung des Setup

Voraussetzung für die Nutzung der Oracle Finanzanwendungen ist die Voreinstellung des Systems. Diese Einrichtung der Hintergrundeinstellungen wird auch *Setup* genannt. Beispiel für bestimmte Einstellungen im Setup ist die Erstellung des Kontoflexfelds. Die Setup-Einstellungen haben verschiedene Auswirkungen auf die Arbeit des Anwenders. So steuern einige Setup-Einstellungen die Funktionsweise des Systems, andere Setup-Schritte führen zur Erstellung von Wertelisten. Zudem hat das Setup die Aufgabe, bestimmte Vorgabewerte zu bestimmen. Wenn man beispielsweise einen Lieferanten anlegt, sind viele Felder schon vorbelegt. Diese Vorgabewerte stammen aus dem Setup.
Der Großteil der Setup-Schritte lässt sich im Menü *Einrichten* des Navigators einsehen. Dieses Menü steht aber nur den Systemverwaltern zur Verfügung, da eine Änderung im Setup Auswirkungen auf das ganze System haben kann.

Teil B: Praktischer Teil

Die praktische Aufgabe dieser Diplomarbeit besteht darin, eine Datenbankanwendung für die Stiftung caesar zu erstellen, welche als Modul in *Oracle Applications* eingebunden wird.

In diesem Kapitel wird die Entwicklung und die Einbindung in Oracle Applications von InnoCon dargestellt.

Beschreibung des Anwendungsproblems

Ist-Zustand

Projektideen werden in der Stiftung caesar in ausgedruckten Excel-Sheets von Hand erfasst. Eine Datenbank für die Speicherung der Ideen, die an die Stiftung caesar herangetragen werden, existiert nicht. Dadurch können die Ideen hinsichtlich ihrer Realisierung nur unzureichend bewertet werden. Auch eine Abfrage sowie die Bewertung von Projektideen ist schwierig.

Soll-Zustand

Projektideen sollen in einer Datenbank gespeichert werden. Da in der Stiftung caesar die Anwendung *Oracle Applications* eingesetzt wird, ist es sinnvoll, die Ideen mit Hilfe dieses Programms zu speichern, und nicht eine neue Datenbank aufzusetzen. Dadurch wird auch die Wartung der Datenbankanwendung gesichert.

Projektideen sollen mit Kurzbeschreibung und Titel erfasst werden. Außerdem müssen Folgedokumente für die Projektideen aufgenommen werden können. Danach wird die Idee einer Vorselektion unterzogen, in der entschieden wird, ob die Idee weiterverfolgt oder nicht weiterverfolgt wird. Um den Projektverlauf zu dokumentieren, müssen alle Projektdokumente wie der Projektantrag oder das Projektgutachten in der Datenbank gespeichert werden.

7. Eigene Anwendung in Oracle Applications erstellen

Um eigene Anwendungen in Oracle Applications erstellen zu können, sollte man nach Angabe von Oracle wie folgt vorgehen[43]:

1. Verzeichnisstruktur auf dem Server erstellen
2. Umgebungsvariable anpassen
3. Tablespaces erstellen
4. Eigene Anwendung registrieren
5. Datenbankbenutzter (Zuständigkeit) für die eigene Anwendung erstellen und in Oracle Applications registrieren
6. Zuständigkeit einer Datengruppe zuordnen
7. Tabellen erstellen und registrieren
8. Masken mit Forms-Builder 4.5 erstellen und in Oracle Applications registrieren
9. Reports erstellen und in Oracle Applications registrieren
10. Menübäume erstellen
11. Zugriff für Anwender von Oracle Applications erteilen

Im folgenden wird die Einbindung einer eigenen Anwendung in Oracle Applications dokumentiert, wobei folgende Abkürzungen verwendet werden:

NAV: ist die Abkürzung für Navigation. Hier wird die Navigation in Oracle Applications dargestellt, unter welcher Zuständigkeit und im welchem Menüpunkt (Maske) die Anpassungen (Setup) vorgenommen werden.

ERP-Server: hierbei ist der Server gemeint, auf dem die Datenbank für Oracle Applications aufgesetzt wurde.

[43] Vgl.: „Oracle Applications Develper's Guide" Release 11, (1998) Oracle Press.

7.1 Verzeichnisstrukturen anlegen

Der Aufbau der Verzeichnisstruktur wird während der Installation von *Oracle Applications* vorgenommen. Den Basispfad auf dem ERP-Server, in dem *Oracle Applications* installiert wurde, bezeichnet man als **APPL_TOP**.
Jede Datenbank-Zuständigkeit in *Oracle Applications* erhält ein eigenes lokales Verzeichnis auf dem ERP-Server, beispielsweise wird der Zuständigkeit CE (CE steht für Cash Management) das Verzeichnis auf dem ERP-Server „erp0/local/d06/applmgr/DEV/ce“ zugeordnet. Wenn weitere Unterverzeichnisse für eigene Anwendungen geschaffen werden, muss die vorgegebene Verzeichnisstruktur übernommen werden.
Auf dem ERP-Server wird im Verzeichnis APPL_TOP ein Unterverzeichnis „InnoCon/1.0.0“ angelegt, welches das INNOCON_TOP darstellen soll. Dies ist der Basispfad für die Anwendung InnoCon, in dem Masken und Reports gespeichert werden.
Das Standardverzeichnis für InnoCon ist:

/local/d06/applmgr/DEV/innocon/1.0.0

Zusätzlich werden hier Unterverzeichnisse angelegt. Dort werden die benutzerdefinierten Masken, Reports sowie der SQL-Code gespeichert.

Verzeichnisse für Forms-Masken:

/local/d06/applmgr/DEV/innocon/1.0.0/forms/D/

Verzeichnis für Reports:

/local/d06/applmgr/DEV/innocon/1.0.0/reports

Die Verzeichnisstruktur wurde mit Zugriff auf den ERP-Server über Telnet erstellt. Es muss sichergestellt werden, dass der Zugriff für die Verzeichnisse freigegeben wird. Dies kann mit dem Unix-Befehl “chmod 750“ auf dem ERP-Server für das Verzeichnis geändert werden.

7.2 Umgebungsvariable anpassen

Nachdem die Verzeichnisstruktur erstellt wurde, muss die Umgebungsvariable „DEV.env“ um den INNOCON_TOP Eintrag erweitert werden. In der Umgebungsvariablen „DEV.env“ werden die Pfade für die einzelnen Zuständigkeiten auf dem ERP-Server gesetzt.

Auszug aus der DEV.env:

```
CE_TOP="/local/d06/applmgr/DEV/ce/11.0.28"
export CE_TOP
# Anwendung für die Diplomarbeit InnoCon - Johannes Schmidt -
INNOCON_TOP="/local/d06/applmgr/DEV/innocon/1.0.0"
export INNOCON_TOP
```

Mit **INNOCON_TOP** wird das Verzeichnis für die Anwendung InnoCon aufgerufen.

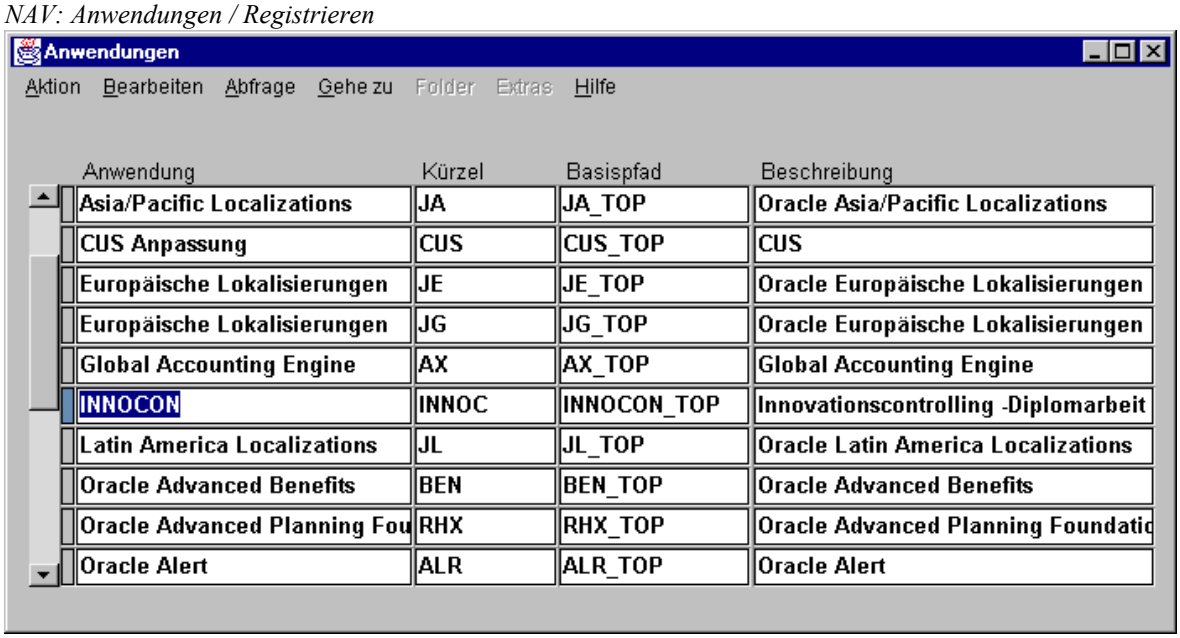
7.3 Tablespaces erstellen

Tablespace ist die logische Zusammenfassung von physikalischen Datendateien. In einem Tablespace können beliebige Segmente (Index-, Rollback-Segmente) gespeichert werden. Für jedes einzelne Segment wird bestimmt, in welchem Tablespace es gespeichert werden soll. Daten, die inhaltlich zueinander gehören, sollen in einem Tablespace gespeichert werden. Aus diesem Grund wird ein Tablespace für die Anwendung InnoCon angelegt. Dieses trägt die Bezeichnung: „*innocd*“.
Um Tablespaces anzulegen, startet man im Oracle Enterprise Manager den Dienst Storage Manager. In diesem werden die Tablespaces verwaltet.

Name des Tablespaces: **INNOCD**
Datendatei : /local/d06/oradata/DEV/INNOC_DATA_01.dbf

Danach wird die Datenbank neu gestartet, damit die Änderungen wirksam werden. Anschließend meldet man sich als Basisbibliothek-Systemadministrator in *Oracle Applications* an und registriert die Anwendung InnoCon wie folgt:

Zuständigkeit: Basisbibliothek-Entwickler
NAV: Anwendungen / Registrieren



Anwendung	Kürzel	Basispfad	Beschreibung
Asia/Pacific Localizations	JA	JA_TOP	Oracle Asia/Pacific Localizations
CUS Anpassung	CUS	CUS_TOP	CUS
Europäische Lokalisierungen	JE	JE_TOP	Oracle Europäische Lokalisierungen
Europäische Lokalisierungen	JG	JG_TOP	Oracle Europäische Lokalisierungen
Global Accounting Engine	AX	AX_TOP	Global Accounting Engine
INNOCON	INNOC	INNOCON_TOP	Innovationscontrolling -Diplomarbeit
Latin America Localizations	JL	JL_TOP	Oracle Latin America Localizations
Oracle Advanced Benefits	BEN	BEN_TOP	Oracle Advanced Benefits
Oracle Advanced Planning Fou	RHX	RHX_TOP	Oracle Advanced Planning Foundatic
Oracle Alert	ALR	ALR_TOP	Oracle Alert

Abbildung 19: Registrierung der Anwendung InnoCon

7.4 Datenbankbenutzter (Zuständigkeit) erstellen

Werden eigene Datenbankobjekte erstellt, sollte auch ein eigener Datenbankbenutzer eingerichtet werden. Dieser sollte alle Rechte an den Superuser APPS weitergeben. Danach muss der Datenbankbenutzer in der Oracle Basisbibliothek registriert werden.

Für die Anwendung InnoCon wird ein Datenbankbenutzer erstellt. Dieser Datenbankbenutzer wird als Zuständigkeit unter *Oracle Applications* bezeichnet. Mit Hilfe dieser Zuständigkeit werden Anwendungen unter InnoCon gestartet.

Der Name des Benutzers für die Anwendung InnoCon lautet *INNOCON*. Zuerst wird mit dem Programm SQL*Plus unter der Kennung des Superuser APPS der Datenbankbenutzer angelegt:

Syntax:

create user INNOCON identified by innocon
default tablespace INNOCD
temporary tablespace TEMP
quota unlimited on INNOCD
quota unlimited on TEMP

Dem Datenbankbenutzer müssen Zugriffsrechte für die Anmeldung an der Datenbank sowie Berechtigungen für das Erstellen von Tabellen, Views und Sequenzen erteilt werden.

Syntax:

GRANT CONNECT TO innocon;
GRANT CREATE TABLE TO innocon;
GRANT CREATE VIEW TO innocon;
GRANT CREATE SEQUENCE TO innocon;

Zuständigkeit registrieren

Wie im Kapitel über Oracle Financials erwähnt (vgl. Kapitel 6: Oracle Financials - Ein Überblick), werden einer Zuständigkeit mehrere eigene Module sowie Reports zugeordnet. Die Zuständigkeit (Datenbankbenutzer) INNOCON wird in *Oracle Applications* wie folgt registriert:

Zuständigkeit: Systemadministrator
NAV: Sicherheitsregeln Oracle_Kennungen Registrieren

Oracle Benutzer

Aktion Bearbeiten Abfrage Gehe zu Folder Extras Hilfe

Datenbank Benutzer	Kennwort	Berechtigung	Installation Gruppe	Beschreibung
ENG		Aktiviert	1	Oracle Konstruktionsdaten Account
FA		Aktiviert	0	Oracle Anlagen Account
FLM		Aktiviert	1	Oracle Flow Manufacturing Account
GL		Aktiviert	0	Oracle Rechnungswesen Account
HR		Aktiviert	0	Oracle Personal Account
HXT		Aktiviert	0	Oracle Zeit-Management Account
ICX		Aktiviert	0	Oracle Self-Service Web Applicatio
INNOCON		Aktiviert	1	Innovationscontrolling - Diplomarb
INV		Aktiviert	1	Oracle Lager Account
JE		Aktiviert	1	Europäische Lokalisierungen Accou

Abbildung 20: Datenbankbenutzter INNOCON

Der Datenbankbenutzer INNOCON muss noch einer Datengruppe zugeordnet werden.

7.4.1 Datengruppe

Eine Datengruppe bestimmt die Zuordnung zwischen den Produkten von *Oracle Applications* und Oracle-Kennungen (Zuständigkeiten). Eine Datengruppe legt fest, mit welchem Datenbank-Account die Masken, Hintergrundprogramme und Auswertungen einer Zuständigkeit verbunden werden.

Bei der Installation werden Datengruppen in *Oracle Applications* automatisch definiert. Es gibt eine Datengruppe mit dem Namen „STANDARD“ in der alle Oracle Anwendungen mit der Oracle-Kennung APPS verbunden werden. Bei der mehrfachen Installation von Oracle Modulen müssen die Datenbankobjekte differenziert werden können. Ein Menübaum ebenso wie ein Report kann nur innerhalb einer Datengruppe auf Daten zugreifen.

In der Abbildung 21 wird die Datengruppe **Standard** um folgenden Eintrag der Anwendung INNOCON erweitert:

Zuständigkeit: Systemadministrator
NAV: Sicherheitsregeln Oracle-Kennungen Datengruppen

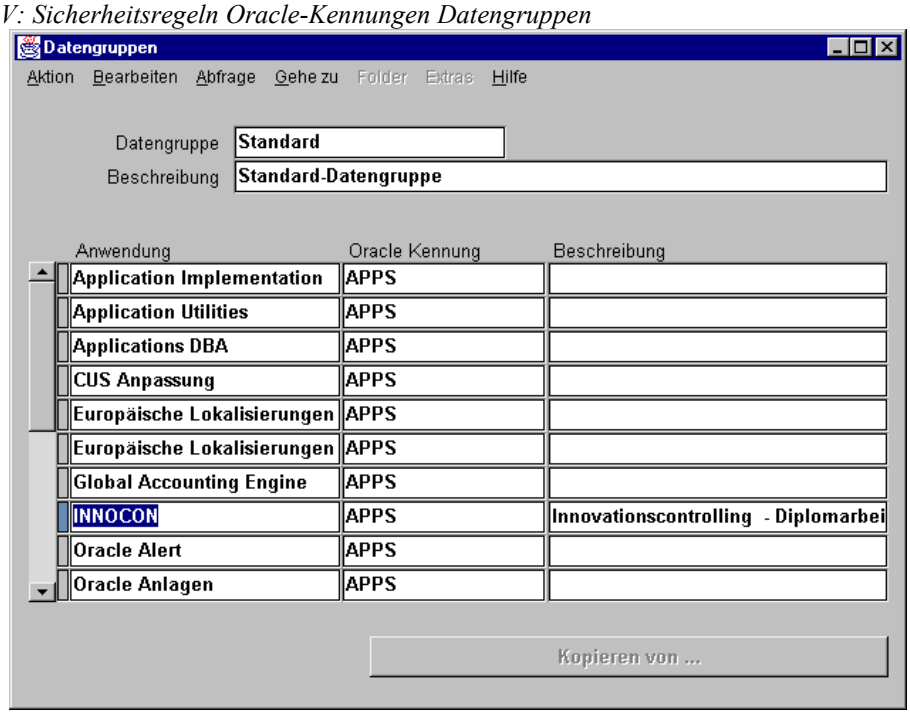


Abbildung 21: Datengruppe erweitern

7.5 Tabellen, Sequenzen erstellen

Aus den Ergebnissen der Analysephase (Pflichtenheft) wurde das ER-Modell der Anwendung InnoCon mit dem Werkzeug *Entity-Relationship-Diagrammer* von *Oracle Designer/2000* erstellt und mit dem *Database-Designer* generiert (vgl. Kapitel 5: Entwicklungswerkzeuge). Die erzeugten SQL-Skripte für die Tabellen wurden mit dem Programm SQL*Plus auf der Datenbank DEV ausgeführt. Alle Tabellenbezeichnungen beginnen mit XX. Dies soll verdeutlichen, dass dies Tabellen der eigenen Anwendung in *Oracle Applications* sind.

Jede Tabelle hat zusätzliche Spalten, die im Datenmodell der Anwendung InnoCon (vgl. Tabelle 5: Attributbeschreibung der Anwendung InnoCon) nicht aufgeführt werden. Diese Spalten bezeichnen das Erstelldatum und das Änderungsdatum sowie von wem der Datensatz der Tabelle erstellt bzw. geändert wurde. In *Oracle Applications* werden diese Spalten als „WHO-Columns“ bezeichnet.

Um die Tabellennamen möglichst kurz zu halten (10 Zeichen) wurden folgende Bezeichnungen für die Tabellen gewählt:

Tabellenname	
XXPRJ	*Projektidee*
XXKP	*Kontaktperson*
XXPRJDOKU	*Projektdokumente*
XXKONT	*Kontakte*
XXFB	*Forschungsbereich*
XXVORS	*Vorselektion*
XXFOL	*Folgedokumente*
XXWV	*Wiedervorlage*

In der folgenden Abbildung (vgl. Abbildung 22: ER-Diagramm der Anwendung InnoCon) wird das ER-Diagramm der Anwendung InnoCon dargestellt.

ER-Modell der Anwendung InnoCon

Abbildung 22: ER-Diagramm der Anwendung InnoCon

Datenmodell der Anwendung InnoCon:

Tabellenname	Spalte	Beschreibung	Typ(Länge)	0	
XXPRJ	XXPRJ_ID	Projektnummer	NUMBER(10)	N	PK
	TITEL	Projekttitel	VARCHAR2(100)	Y	PK
	DATUM	Datum der Ersterfassung	DATE(11)	N	
	BESCHREIBUNG	Projekt Beschreibung	VARCHAR2(1000)	Y	
	GEW_GRUNDLAGENFORSCHUNG	Gewichtungsspalte	NUMBER(20)	*Y*	
	GEW_ANWENDUNG	Gewichtungsspalte	NUMBER(20)	*Y*	
	GEW_ARBEITSMARKT	Gewichtungsspalte	NUMBER(20)	*Y*	
	GEW_PRESTIGE	Gewichtungsspalte	NUMBER(20)	*Y*	
	FK_XXFB_ID	Fremdschlüssel: Forschungsbereich	NUMBER(10)	Y	FK
	FK_XXKONT_ID	Fremdschlüssel: Kontakte	NUMBER(10)	Y	FK
XXKP	XXKP_ID	Kontaktperson ID	NUMBER(10)	N	PK
	ANREDE		VARCHAR2(20)	Y	
	TITEL		VARCHAR2(100)	Y	
	VORNAME		VARCHAR2(100)	Y	
	NAME		VARCHAR2(100)	Y	
	STRASSE		VARCHAR2(50)	Y	
	PLZ		NUMBER(6)	Y	
	ORT		VARCHAR2(50)	Y	
	TELEFON		VARCHAR2(50)	Y	
	EMAIL		VARCHAR2(50)	Y	
	FK_XXPRJ_ID	Fremdschlüssel: Projekt	NUMBER(10)	N	FK
XXPRJDOKU	XXPRJDOKU_ID	Projektdokumenten ID	NUMBER(10)	N	PK
	PLAN_BUDGET		NUMBER(10)	*Y*	
	PLAN_PERSONAL		VARCHAR2(20)	*Y*	
	PRJ_ANTRAG_DATUM		DATE(11)	*Y*	
	PRJ_ANTRAG_RUECK_DATUM		DATE(11)	*Y*	
	PRJ_ANTRAG_DATEIANHANG		VARCHAR2(70)	*Y*	
	BESCHLUS_DATUM		DATE(11)	*Y*	
	BESCHLUS_KURZTEXT		VARCHAR2(200)	*Y*	
	GUTACHTEN_DATUM		DATE(11)	*Y*	
	GUTACHTEN_BESCHREIBUNG		VARCHAR2(200)	*Y*	
	FK_XXPRJ_ID	Fremdschlüssel: Projekt	NUMBER(10)	N	FK
XXKONT	XXKONT_ID	Primärschlüssel	NUMBER(10)	N	PK
	KONTAKTGRUPPE	Auswahl der KGR	VARCHAR2(240)	*Y*	
	SONSTIGE_KGR	Eintrag der KGR	VARCHAR2(240)	*Y*	
	KONTAKTART	Auswahl der KA	VARCHAR2(240)	*Y*	
	SONSTIGE_KA	Eintrag der KA	VARCHAR2(240)	*Y*	
XXFB	XXFB_ID	Primärschlüssel	NUMBER(10)	N	PK
	TRIPLET		VARCHAR2(240)	*Y*	
	DISZIPLIN	Disziplin ist: Betriebswirtschaftlehre Finanzmathematik	VARCHAR2(240)	*Y*	
	THEMENKREIS	Abgrenzung der Disziplin	VARCHAR2(240)	*Y*	
	BEREICH	1. Materialwissenschaften 2. Koppelung elektronischer und biologischer Systeme 3. Kommunikationsergonomie	VARCHAR2(240)	*Y*	
	PROJEKTART	Auswahl der Projektart : Pilotprojekt	VARCHAR2(240)	*Y*	
XXVORS	XXVORS_ID	Primärschlüssel	NUMBER(10)	N	PK
	DATUM		DATE(11)	N	
	BEGRUENDUNG		VARCHAR2(400)	*Y*	
	VORSELEKTIONSBESCHLUSS		VARCHAR2(240)	*Y*	
	FK_XXPRJ_ID	Fremdschlüssel: Projekt	NUMBER(10)	N	
XXFOL	XXFOL_ID	Primärschlüssel	NUMBER(10)	N	PK
	DATUM		DATE(11)	N	
	GESPRAECHSNOTIZ		VARCHAR2(400)	*Y*	
	VEREINBARTE_AKTIVITAETEN		VARCHAR2(400)	*Y*	
	FK_XXPRJ_ID	Fremdschlüssel: Projekt	NUMBER(10)	N	FK

Tabellenname	Spalte	Beschreibung	Typ(Länge)	0	
XXWV	XXWV_ID	Primärschlüssel	NUMBER(10)	N	PK
	DATUM		DATE(11)	N	
	VERTEILER_AN	Weiterleiten der Idee	VARCHAR2(100)	*Y*	
	ANLAGEN	Anlagen zur Idee	VARCHAR2(100)	*Y*	
	BEARBEITER	Bearbeiter der Projektidee	VARCHAR2(100)	*Y*	
	VERSAND_PRJANTRAG		VARCHAR(20)	*Y*	
	Task	z.B.: Informationen einholen	VARCHAR2(100)		
	FK_XXVORS_ID	Fremdschlüssel Vorselektion	NUMBER(10)	*Y*	
	FK_XXFOL_ID	Fremdschlüssel: Folgedokumente	NUMBER(10)	*Y*	
	FK_XXPRJ_ID	Fremdschlüssel: Projektidee	NUMBER(10)	N	

Tabelle 4: Attributbeschreibung der Anwendung InnoCon

In der Praxis ist es häufig so, dass man zusätzliche Tabellen benötigt, die in der Entwurfsphase (ER-Modell) noch nicht spezifiziert wurden. Das ER-Modell wurde um folgende Tabellen für die Projektbewertung erweitert:

Tabellenname	Beschreibung
XXBEW_DURCH	Bewertung der Durchführbarkeit
XXBEW_EIG_WISS	Eigene wissenschaftliche Stärke
XXBEW_RISIKO	Bewertung des Risikos
XXBEW_WIRT	wirtschaftliche Attraktivität
XXBEW_WISS	wissenschaftliche Attraktivität

Tabelle 5: Beschreibung der Projektbewertungstabellen

Datenmodell der erweiterten Tabellen:

Tabellenname	Spalte	Beschreibung	Typ(Länge)	0	Schlüssel
XBEW_DURCH	XXBEW_DURCH_ID	Primärschlüssel	NUMBER(10)	N	PK
	FK_XXPRJ_ID	Fremdschlüssel: Projektidee	NUMBER(10)	N	FK
	DURCH_1	Budgetvolumen	VARCHAR2(10)	*Y*	
	DURCH_2	Freie Kapazitäten	VARCHAR2(10)	*Y*	
	DURCH_3	Verfügbarkeit von Human Ressources	VARCHAR2(10)	*Y*	
	DURCH_4	Verfügbarkeit der Technologie	VARCHAR2(10)	*Y*	
	DURCH_5	Logistischer Rahmen	VARCHAR2(10)	*Y*	
	DURCH_6	Ethische Beurteilung	VARCHAR2(10)	*Y*	
	DURCH_7	Rechtliche Durchführbarkeit	VARCHAR2(10)	*Y*	
XXBEW_EIG_WISS	XXBEW_EIG_WISS_ID	Primärschlüssel	NUMBER(10)	N	PK
	FK_XXPRJ_ID	Fremdschlüssel: Projektidee	NUMBER(10)	N	FK
	WISS_EIG_GEW_1	Gewichtung (1-15): Human Capital	NUMBER(10)	Y	
	WISS_EIG_1	Gewichtungsfaktor: Human Capital	NUMBER(10)	Y	
	WISS_SUM_EIG_1	Summe: Human Capital	NUMBER(10)	Y	
	WISS_EIG_GEW_2	Gewichtung (1-15): Technologie	NUMBER(10)	Y	
	WISS_EIG_2	Gewichtungsfaktor: Technologie	NUMBER(10)	Y	
	WISS_SUM_EIG_2	Summe: Technologie	NUMBER(10)	Y	
	SUMME_EIG	Summe der Gewichtungsfaktoren	NUMBER(10)	Y	
	SUMME_EIG_SUM	Beitrag zur Gesamtattraktivität	NUMBER(10)	Y	
XXBEW_RISIKO	XXBEW_RISIKO_ID	Primärschlüssel	NUMBER(10)	N	PK

Tabellenname	Spalte	Beschreibung	Typ(Länge)	0	Schlüssel
	FK_XXPRJ_ID	Fremdschlüssel: Projektidee	NUMBER(10)	N	FK
	RISIKO_GEW_1	Gewichtung (1-15): Änderung der Nachfrage	NUMBER(10)	Y	
	RISIKO_1	Gewichtungsfaktor: Änderung der Nachfrage	NUMBER(10)	Y	
	RISIKO_SUM_1	Summe: Änderung der Nachfrage	NUMBER(10)	Y	
	RISIKO_PRO_1	Prozentuale Verteilung Änderung der Nachfrage	NUMBER(10,3)	Y	
	RISIKO_GEW_2	Gewichtung (1-15): Ergebniserzielung in der Planperiode	NUMBER(10)	Y	
	RISIKO_2	Gewichtungsfaktor: Ergebniserzielung in der Planperiode	NUMBER(10)	Y	
	RISIKO_SUM_2	Summe: Ergebniserzielung in der Planperiode	NUMBER(10)	Y	
	RISIKO_PRO_2	Prozentuale Verteilung Ergebniserzielung in der Planperiode	NUMBER(10,3)	Y	
	RISIKO_GEW_3	Gewichtung (1-15): Internationale Konkurrenz	NUMBER(10)	Y	
	RISIKO_3	Gewichtungsfaktor: Internationale Konkurrenz	NUMBER(10)	Y	
	RISIKO_SUM_3	Summe: Internationale Konkurrenz	NUMBER(10)	Y	
	RISIKO_PRO_3	Prozentuale Verteilung Internationale Konkurrenz	NUMBER(10,3)	Y	
	RISIKO_GEW_4	Gewichtung (1-15): Innovationsgeschwindigkeit	NUMBER(10)	Y	
	RISIKO_4	Gewichtungsfaktor: Innovationsgeschwindigkeit	NUMBER(10)	Y	
	RISIKO_SUM_4	Summe: Innovationsgeschwindigkeit	NUMBER(10)	Y	
	RISIKO_PRO_4	Prozentuale Verteilung Innovationsgeschwindigkeit	NUMBER(10,3)	Y	
	RISIKO_GEW_5	Gewichtung (1-15): Nationale Konkurrenz	NUMBER(10)	Y	
	RISIKO_5	Gewichtungsfaktor: Nationale Konkurrenz	NUMBER(10)	Y	
	RISIKO_SUM_5	Summe: Nationale Konkurrenz	NUMBER(10)	Y	
	RISIKO_PRO_5	Prozentuale Verteilung Nationale Konkurrenz	NUMBER(10,3)	Y	
	SUMME_RISIKO	Summe der Gewichtungsfaktoren	NUMBER(10)	Y	
	SUMME_RISIKO_SUM	Beitrag zur Gesamtattraktivität	NUMBER(10)	Y	
XXBEW_WIRT	XXBEW_WIRT_ID	Primärschlüssel	NUMBER(10)	N	PK
	FK_XXPRJ_ID	Fremdschlüssel: Projektidee	NUMBER(10)	N	FK
	WIRT_GEW_1	Gewichtungsfaktor: Anwendungsnähe	NUMBER(10)	Y	
	WIRT_1	Gewichtung (1-15): Anwendungsnähe	NUMBER(10)	Y	
	WIRT_SUM_1	Summe: Anwendungsnähe	NUMBER(10)	Y	
	WIRT_GEW_2	Gewichtungsfaktor: Patentierbarkeit	NUMBER(10)	Y	
	WIRT_2	Gewichtung (1-15): Patentierbarkeit	NUMBER(10)	Y	
	WIRT_SUM_2	Summe: Patentierbarkeit	NUMBER(10)	Y	
	WIRT_GEW_3	Gewichtungsfaktor: Eigene Nutzbarkeit	NUMBER(10)	Y	
	WIRT_3	Gewichtung (1-15): Eigene Nutzbarkeit	NUMBER(10)	Y	
	WIRT_SUM_3	Summe: Eigene Nutzbarkeit	NUMBER(10)	Y	
	WIRT_GEW_4	Gewichtungsfaktor: Nachfrage	NUMBER(10)	Y	
	WIRT_4	Gewichtung (1-15): Nachfrage	NUMBER(10)	Y	
	WIRT_SUM_4	Summe: Nachfrage	NUMBER(10)	Y	
	WIRT_GEW_5	Gewichtungsfaktor: Kapitalintensität	NUMBER(10)	Y	
	WIRT_5	Gewichtung (1-15): Kapitalintensität	NUMBER(10)	Y	
	WIRT_SUM_5	Summe: Kapitalintensität	NUMBER(10)	Y	
	WIRT_GEW_6	Gewichtungsfaktor: Technologieintensität	NUMBER(10)	Y	

Tabellenname	Spalte	Beschreibung	Typ(Länge)	0	Schlüssel
	WIRT_6	Gewichtung (1-15): Technologieintensität	NUMBER(10)	Y	
	WIRT_SUM_6	Summe: Technologieintensität	NUMBER(10)	Y	
	WIRT_GEW_7	Gewichtungsfaktor: Personalintensität	NUMBER(10)	Y	
	WIRT_7	Gewichtung (1-15): Personalintensität	NUMBER(10)	Y	
	WIRT_SUM_7	Summe: Personalintensität	NUMBER(10)	Y	
	WIRT_GEW_8	Gewichtungsfaktor: Zukünftige Nutzbarkeit Human Capital	NUMBER(10)	Y	
	WIRT_8	Gewichtung (1-15): Zukünftige Nutzbarkeit Human Capital	NUMBER(10)	Y	
	WIRT_SUM_8	Summe: Zukünftige Nutzbarkeit Human Capital	NUMBER(10)	Y	
	WIRT_GEW_9	Gewichtungsfaktor: Zukünftige Nutzbarkeit der Technologie	NUMBER(10)	Y	
	WIRT_9	Gewichtung (1-15): Zukünftige Nutzbarkeit der Technologie	NUMBER(10)	Y	
	WIRT_SUM_9	Summe: Zukünftige Nutzbarkeit der Technologie	NUMBER(10)	Y	
	WIRT_GEW_10	Gewichtungsfaktor: Vorteilhaftigkeit für die Region	NUMBER(10)	Y	
	WIRT_10	Gewichtung (1-15): Vorteilhaftigkeit für die Region	NUMBER(10)	Y	
	WIRT_SUM_10	Summe: Vorteilhaftigkeit für die Region	NUMBER(10)	Y	
	SUMME_WIRT	Summe der Gewichtungsfaktoren	NUMBER(10,3)	N	
	SUMME_WIRT_SUM	Beitrag zur Gesamtattraktivität	NUMBER(10,3)	N	
XXBEW_WISS	XXBEW_WISS_ID	Primärschlüssel	NUMBER(10)	N	PK
	FK_XXPRJ_ID	Fremdschlüssel: Projektidee	NUMBER(10)	N	FK
	WISS_GEW_1	Gewichtungsfaktor: Rang, Reputation	NUMBER(10)	Y	
	WISS_1	Gewichtung (1-15): Rang, Reputation	NUMBER(10)	Y	
	WISS_SUM_1	Summe: Rang, Reputation	NUMBER(10)	Y	
	WISS_GEW_2	Gewichtungsfaktor: Entwicklungsmöglichkeiten	NUMBER(10)	Y	
	WISS_2	Gewichtung (1-15): Entwicklungsmöglichkeiten	NUMBER(10)	Y	
	WISS_SUM_2	Summe: Entwicklungsmöglichkeiten	NUMBER(10)	Y	
	WISS_GEW_3	Gewichtungsfaktor: Bedeutung international	NUMBER(10)	Y	
	WISS_3	Gewichtung (1-15): Bedeutung international	NUMBER(10)	Y	
	WISS_SUM_3	Summe: Bedeutung international	NUMBER(10)	Y	
	WISS_GEW_4	Gewichtungsfaktor: Bedeutung international	NUMBER(10)	Y	
	WISS_4	Gewichtung (1-15): Bedeutung national	NUMBER(10)	Y	
	WISS_SUM_4	Summe: Bedeutung national	NUMBER(10)	Y	
	WISS_GEW_5	Gewichtungsfaktor: Wissenschaftsfortschritt international	NUMBER(10)	Y	
	WISS_5	Gewichtung (1-15): Wissenschaftsfortschritt international	NUMBER(10)	Y	
	WISS_SUM_5	Summe: Wissenschaftsfortschritt international	NUMBER(10)	Y	
	WISS_GEW_6	Gewichtungsfaktor: Wissenschaftsfortschritt national	NUMBER(10)	Y	
	WISS_6	Gewichtung (1-15): Wissenschaftsfortschritt national	NUMBER(10)	Y	
	WISS_SUM_6	Summe: Wissenschaftsfortschritt national	NUMBER(10)	Y	
	WISS_GEW_7	Gewichtungsfaktor: Verfügbarkeit Technologien	NUMBER(10)	Y	
	WISS_7	Gewichtung (1-15): Verfügbarkeit Technologien	NUMBER(10)	Y	
	WISS_SUM_7	Summe: Verfügbarkeit Technologien	NUMBER(10)	Y	

Tabellenname	Spalte	Beschreibung	Typ(Länge)	0	Schlüssel
	WISS_GEW_8	Gewichtungsfaktor: Verfügbarkeit Human Ressources	NUMBER(10)	Y	
	WISS_8	Gewichtung (1-15): Verfügbarkeit Human Ressources	NUMBER(10)	Y	
	WISS_SUM_8	Summe: Verfügbarkeit Human Ressources	NUMBER(10)	Y	
	WISS_GEW_9	Gewichtungsfaktor: Möglichkeit des Know-how Importes	NUMBER(10)	Y	
	WISS_9	Gewichtung (1-15): Möglichkeit des Know-how Importes	NUMBER(10)	Y	
	WISS_SUM_9	Summe: Möglichkeit des Know-how Importes	NUMBER(10)	Y	
	WISS_GEW_10	Gewichtungsfaktor: Forschungsaufwand	NUMBER(10)	Y	
	WISS_10	Gewichtung (1-15): Forschungsaufwand	NUMBER(10)	Y	
	WISS_SUM_10	Summe: Forschungsaufwand	NUMBER(10)	Y	
	SUMME_WISS	Summe der Gewichtungsfaktoren	NUMBER(10,3)		
	SUMME_WISS_SUM	Beitrag zur Gesamtattraktivität	NUMBER(10,3)		

Tabelle 6: Attributbeschreibung der Projektbewerungstabellen

Zugriffsrechte

Dem Superuser APPS müssen für alle angelegten Datenbankobjekte Zugriffsrechte erteilt werden. Dies erfolgt nach der Anmeldung des Datenbankbenutzers INNOCON mit folgender Syntax:

grant all on „TABELLENNAME DES BENUTZERS“ to APPS;

Beispiel für die Tabelle Projektideen:

grant all on XXPRJ to APPS;

(Auf die genaue Syntax für die Anwendung InnoCon wird in Kapitel 9 eingegangen).

7.5.1 Tabellen in *Oracle Applications* registrieren (AD_DD.register)

Nachdem alle benötigen Tabellen erstellt wurden, ist es notwendig, das Package AD_DD des Datenbankbenutzer APPS auszuführen. Dieses Package registriert die Benutzer-Tabellen sowie die Spalten in *Oracle Applications*.

Nachdem man sich mit der Kennung APPS unter SQL*PLUS an der Datenbank angemeldet hat, führt man einen EXECUTE Befehl für die Prozedur AD_DD.register aus, in der die Informationen über den Aufbau der Tabellen und der Spalten enthalten sind.

Die Prozedur AD_DD ist wie folgt aufgebaut:

1. AD_DD.register_table
Syntax:
execute ad_dd.register_table('INNOC','XXPRJ','T',8,10,90);

Kurzbezeichnung der Anwendung = 'INNOC',

Tabellen Bezeichnung = 'XXPRJ',

Tabellen Typ = 'T',

next extent = 8, % frei =10, % benutzt=90

2. AD_DD.register_column

execute AD_DD.register_column('INNOC', 'XXPRJ','DATUM','2','DATE','11','N','N');

Erläuterung zu AD_DD.register_column:

Feld	*Erläuterung*
'INNOC'	*Kurzbezeichnung der Anwendung*
'XXPRJ'	*Tabellenname*
'DATUM'	*Spaltenname*
'2'	*Reihenfolge der Spalten*
'DATUM'	*Spaltentyp*
'11'	*Spaltenbreite*
'N'	*Nullwerte zulässig, hier würde ein 'Y' eintragen werden, wenn dies eine Pflichtspalte wäre*
'N'	*Translate*

Tabelle 7: Erläuterung zu AD_DD.register_column

Zusätzlich zu den Tabellen sowie Spalten werden Primärschlüssel registriert:

Syntax:

execute ad_dd.register_primary_key_column ('INNOC', 'XXPRJ_ID', 'XXPRJ', 'XXPRJ_ID', '1');

In *Oracle Applications* können die registrieren Tabellen überprüft werden, wie die folgende Abbildung für die Tabelle XXPRJ (Projektideen) zeigt:

Zuständigkeit: Basisbibliothek-Entwickler
NAV: Anwendungen Datenbanken Tabellen

Tabellen

Aktion Bearbeiten Abfrage Gehe zu Folder Extras Hilfe

Tabelle XXPRJ
Anwendung INNOCON
Benutzertabelle XXPRJ
Art Transaktionsdaten
Beschreibung Projektidee

Erweiterung
Ursprüngliche 4
Nächste 8
Minimale 1
Maximale 50

Speicher
Frei 10 %
Belegt 90 %
Erhöhen 0 %

Transaktionen
Ursprüngliche 1
Max Transaktionen 255
Automatisch

Indizes | Primäre Schlüssel | Fremdschlüssel

Folge	Spalte	Benutzerspalte	Spaltenart	Breite	Nac
1	XXPRJ_ID	XXPRJ_ID	Nummer	10	
2	DATUM	DATUM	Datum	11	
3	TITEL	TITEL	Varchar2	100	
4	BESCHREIBUNG	BESCHREIBUNG	Varchar2	1000	
5	GESPRAECHSNOTIZ_ERS	GESPRAECHSNOTIZ_ERSTDOK	Varchar2	1000	

Abbildung 23: Registrierte Tabelle in *Oracle Applications*

Löschen der registrierten Tabellen und Spalten

Wenn die Tabelle oder verschiedene Spalten aus *Oracle Applications* gelöscht werden sollen, so wird folgende Funktionen der Package AD_DD aufgerufen:

Syntax zum löschen des Primärschlüssels:

execute ad_dd.delete_primary_key_column ('INNOC', ' PK_XXPRJ','XXPRJ','0');

Syntax, um eine Spalte zu löschen:

execute ad_dd.delete_column('INNOC', 'XXPRJ','XXPRJ_ID');

Syntax, um die Tabelle zu löschen:

execute ad_dd.delete_table ('INNOC', 'XXPRJ');

Danach erst wird der SQL Befehl *DROP TABLE* für das physikalische Löschen der Tabelle ausgeführt.

7.6 Forms-Masken mit Developer/2000 4.5 erstellen

Alle benutzerdefinierten Masken sollten mit 'XX_' + Maskenname beginnen (wie auch die Tabellen), damit es in *Oracle Applications* zu keiner Verwechslung der Masken kommt.

Um Masken in *Oracle Applications* zu erstellen, wird das Template.fmb in das lokale Verzeichnis des Clients, auf dem die Forms-Masken entwickelt werden, kopiert. Dies ist eine Vorlage zur Erstellung von Masken in *Oracle Applications*.

Die FORMS45_PATH Variable sollte auf das Arbeitsverzeichnis des ERP-Servers gesetzt werden. Dadurch werden die *.pll- und die *.plx-Bibliotheken, die für *Oracle Applications* relevant sind, verwendet. Falls die PATH_VARIABLE nicht gesetzt werden kann, kopiert man die nötigen Bibliotheken in das lokale Forms-Verzeichnis. Dazu kann auf dem lokalen (Client) PC ein Verzeichnis angelegt werden, in das die Bibliotheken kopiert werden.

Nachdem die Forms-Masken (*.fmb Dateien) mit dem *Forms-Designer* erstellt wurden, müssen diese in das Verzeichnis:

/local/d06/applmgr/DEV/au/11.0.28/forms/D

des ERP-Servers kopiert werden. In dem Verzeichnisse au/11.0.28/forms/D sind alle Oracle Applications - Masken als .fmb-Dateien gespeichert. Die Masken werden aus diesem Verzeichnis heraus kompiliert und in das jeweilige Unterverzeichnis als .fmx-Datei kopiert.

Um die Maske auf dem Server zu kompilieren, wird folgende Syntax aufgerufen:

Syntax:

f45gen module=XX_PRJ.fmb userid=innocon/innocon
output_file=/local/d06/applmgr/DEV/innocon/1.0.0/forms/XX_PRJ.fmx
module_type=form batch=no compile_all=special

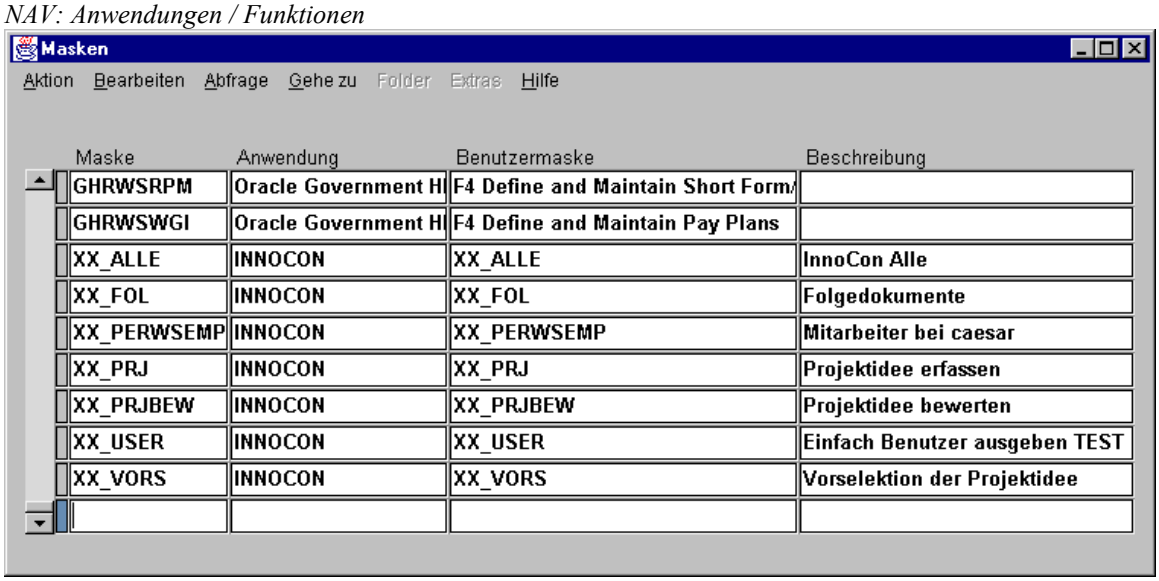
7.6.1 Forms-Masken registrieren

Nachdem die Forms-Maske erstellt wurde, wird diese auf dem ERP-Server kompiliert und in *Oracle Applications* eingetragen bzw. registriert.

Im folgenden wird gezeigt, wie die erstellten Forms-Masken in *Oracle Applications* eingetragen werden.

In der folgenden Maske trägt man unter der Funktion die erstellten Forms-Masken ein, damit in *Oracle Applications* darauf Zugriff genommen werden kann.

Zuständigkeit: Basisbibliothek – Systemadministrator
NAV: Anwendungen / Funktionen



Maske	Anwendung	Benutzermaske	Beschreibung
GHRWSRPM	Oracle Government HI	F4 Define and Maintain Short Form/	
GHRWSWGI	Oracle Government HI	F4 Define and Maintain Pay Plans	
XX_ALLE	INNOCON	XX_ALLE	InnoCon Alle
XX_FOL	INNOCON	XX_FOL	Folgedokumente
XX_PERWSEMP	INNOCON	XX_PERWSEMP	Mitarbeiter bei caesar
XX_PRJ	INNOCON	XX_PRJ	Projektidee erfassen
XX_PRJBEW	INNOCON	XX_PRJBEW	Projektidee bewerten
XX_USER	INNOCON	XX_USER	Einfach Benutzer ausgeben TEST
XX_VORS	INNOCON	XX_VORS	Vorselektion der Projektidee

Abbildung 24: Maskenfunktionen

7.6.2 Reports in *Oracle Applications* registrieren

Wie im Kapitel „Entwicklungswerkzeuge“ dargestellt, werden Reports mit dem *Reports-Designer* entwickelt. Nachdem diese fertiggestellt und in das ERP-Verzeichniss kopiert worden sind, müssen sie in *Oracle Applications* registriert werden. Zuerst wird der Report als ausführbares Hintergrundprogramm definiert und der Anwendung INNOCON zugeordet (vgl. Abbildung 25: Ausführbare Hintergrundprogramme). Diese Eingaben müssen für jeden erstellten Report erfolgen, hier wird jedoch nur die Einstellungen für den Report „Erstdokumentation ausgeben“ (innocon_XXPRJ) dargestellt.

Zuständigkeit: Basisbibliothek-Entwickler
NAV: Hintergrundverarbeitung Ausführbare Hintergrundprogramme

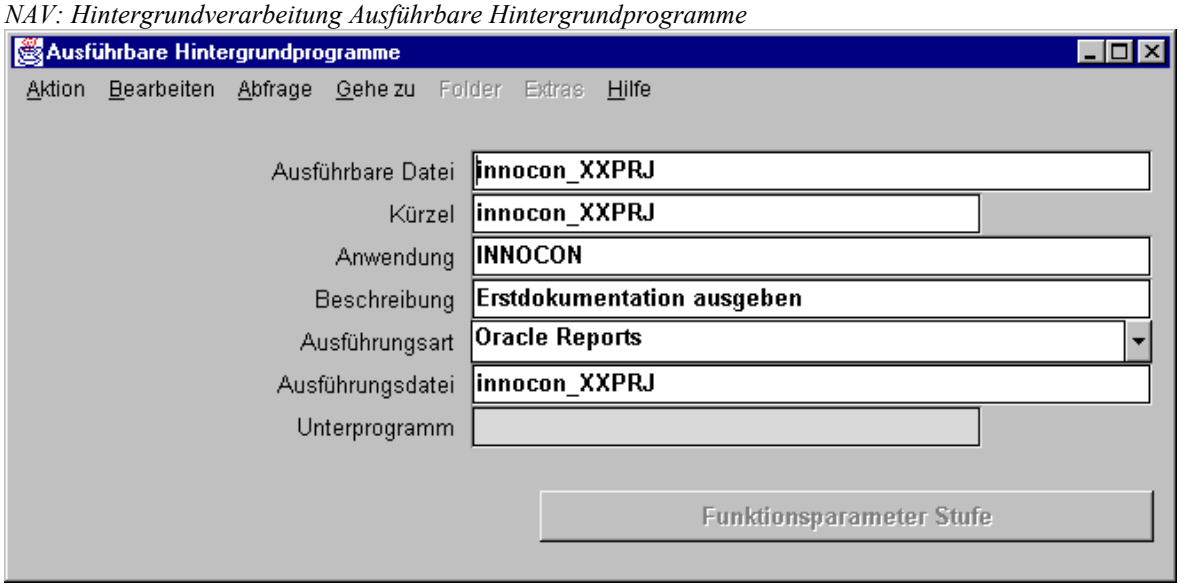


Abbildung 25: Ausführbare Hintergrundprogramme

In der folgenden Maske werden Hintergrund-Programme mit ihrem eigentlichen Programmnamen (z.B. Report INNOCON_XXPRJ) registriert und damit dem System bekanntgemacht.

Durch die Verbindung des Reports mit einer registrierten Anwendung findet das System die ausführbare Datei auf der Seite des Betriebssystems (Basispfad).

Diese so eingebundenen ausführbaren Programme werden dann entweder in eine Bibliothek oder als Hintergrundprozess eingebunden.

Zuständigkeit: Basisbibliothek-Entwickler
NAV: Hintergrundverarbeitung Bibliothek

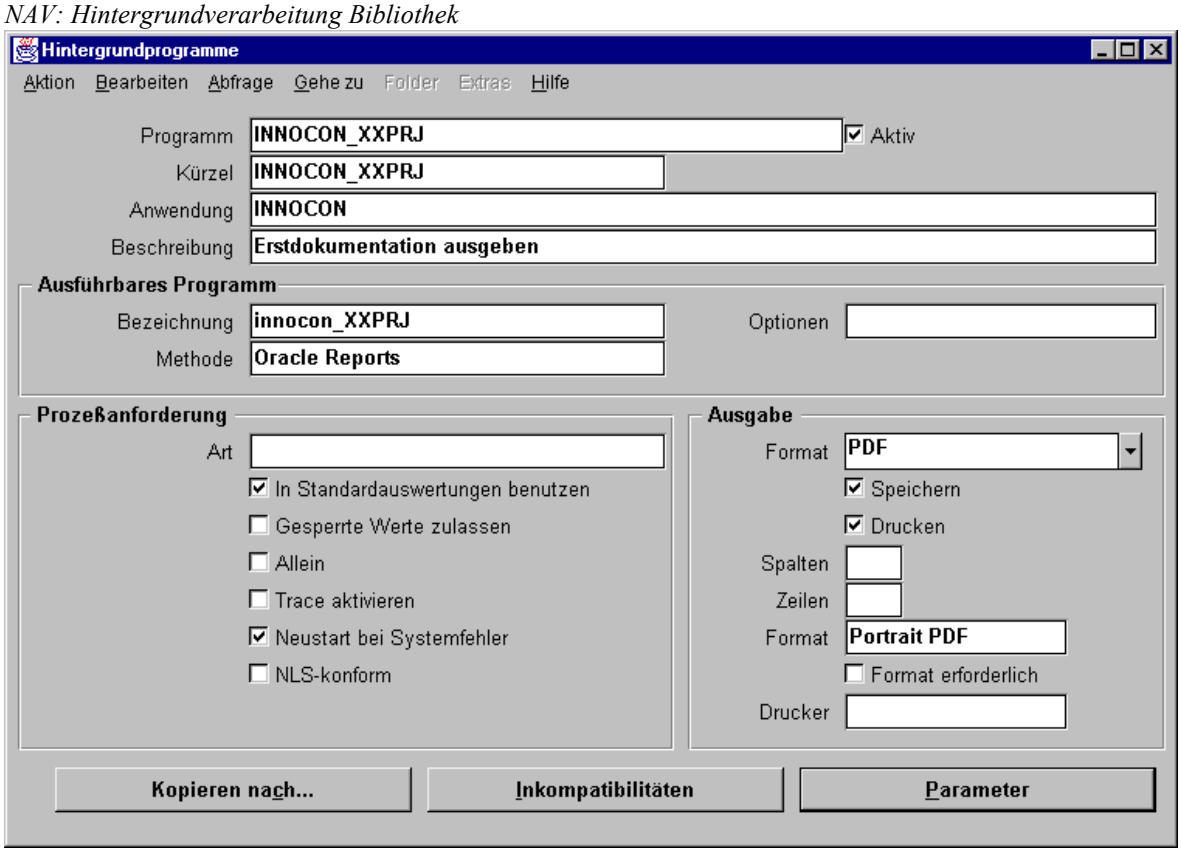


Abbildung 26: Hintergrundprogramme (Reports) registrieren

7.6.3 WerteSet erstellen

Beim Erstellen eines Hintergrundprozesses (Reports) sollte der Anwender nach Auswahl der Projektnummer einen Bericht für diese Projektnummer erzeugen können.

Dazu wird zuerst die Binde-Referenz im Report-Builder erstellt, danach muss man in *Oracle Applications* einen WerteSet erstellen, in dem die Projektidee anhand der Projektnummer ausgewählt werden kann (vgl. Abbildung 27: WerteSet für die Projektnummer).

Zuständigkeit: Basisbibliothek-Entwickler
NAV: Hintergrundverarbeitung Bibliothek

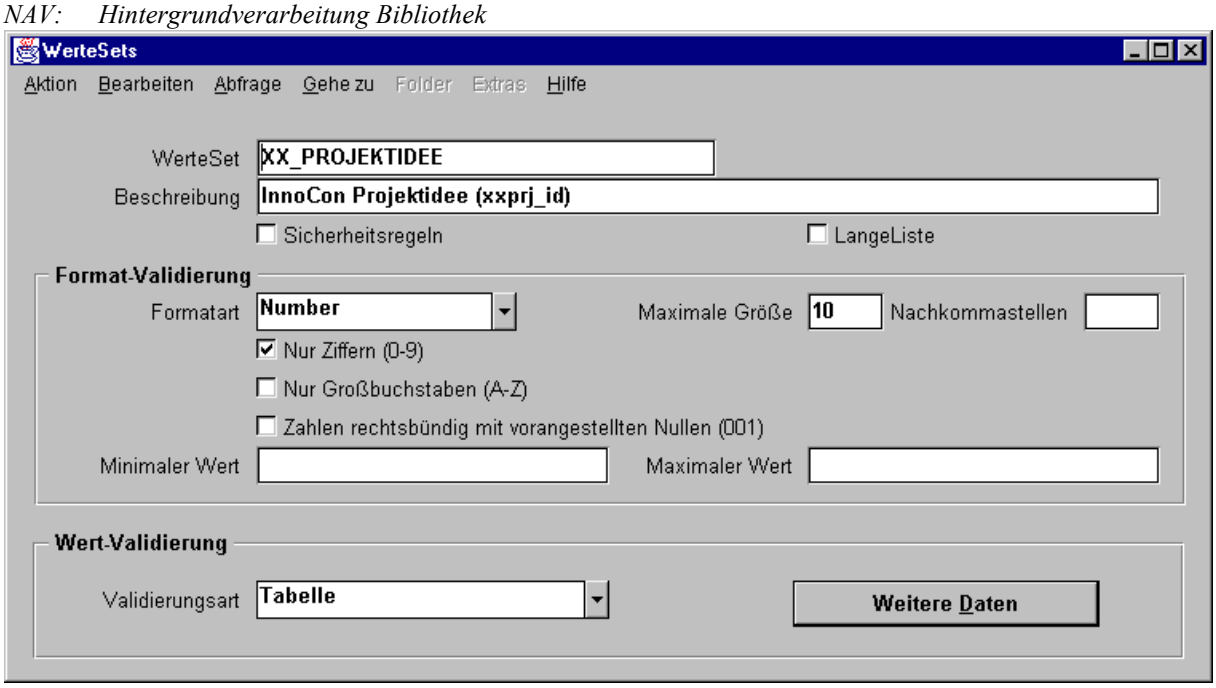


Abbildung 27: WerteSet für die Projektnummer

Zusätzlich zu dem Werte Set muss die Wert-Validierung eingestellt werden. Nach Betätigung der Schaltfläche „Weitere Daten“ gelangt man zu der folgenden Maske, die in Abbildung 28: dargestellt wird.

In der Maske „Validierungstabelle“ der folgenden Abbildung wird die Tabelle XXPRJ (Projektideen) der Anwendung INNOCON eingetragen. Im Sektor Tabellenspalten wird der Wert XXPRJ_ID (Projektnummer) und die Beschreibung des Projekttitels ausgewählt. Zusätlich können hier verschieden SQL-Statements in dem WHERE/ORDER BY-Feld eingetragen werden.

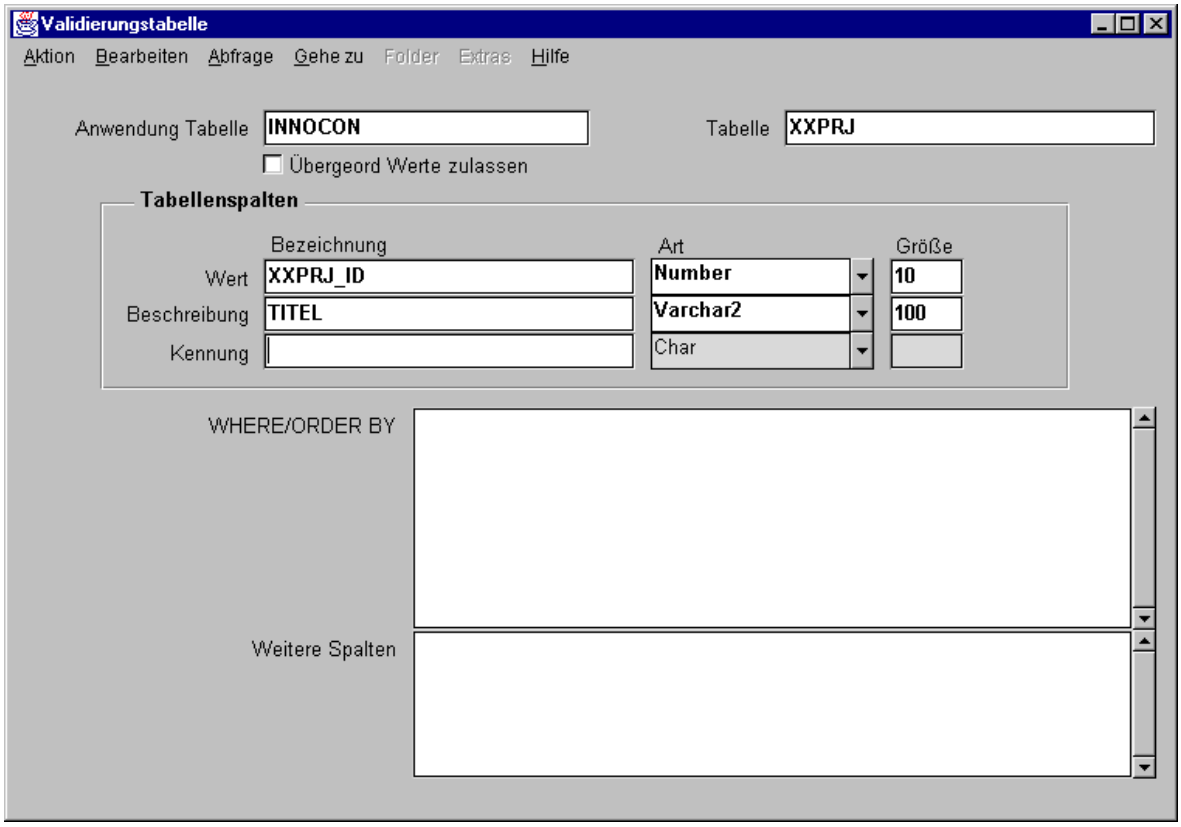


Abbildung 28: WerteSet - Validierungstabelle

7.6.4 Prozessgruppen

Prozessgruppen werden angelegt, um eigene Prozesse starten zu können. Jede Zuständigkeit hat ihre eigene Prozessgruppen. In der Anwendung InnoCon gehören zu den Prozessgruppen Reports, wie die Ausgabe der Erstdokumentation, der Folgedokumente sowie die Auswertungstools.

Zuständigkeit: Basisbibliothek-Entwickler
NAV: Sicherheitsregeln Zuständigkeiten Hintergrundprozesse

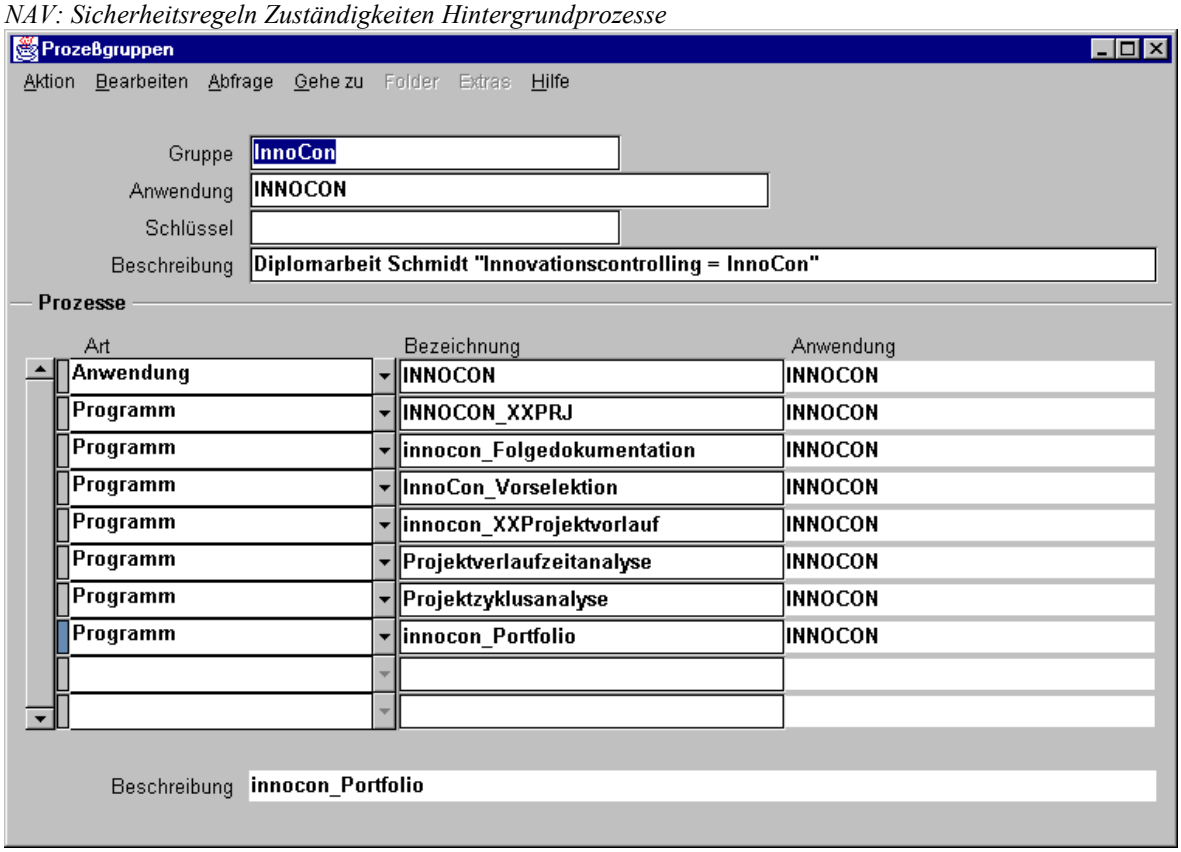


Abbildung 29: Prozessgruppen

7.7 Menübaum erstellen

Ein Menübaum wird immer von unten nach oben aufgebaut, d.h. ein untergeordneter Menübaum kann in einen anderen eingebunden werden.
Dabei entsteht keine exklusive Beziehung. Vielmehr kann ein Menübaum in weiteren Menüs verwendet werden. Bei der Definition der Zuständigkeiten können dann Menüs, Funktionen und Subfunktionen exkludiert werden. In der folgenden Abbildung (vgl. Abbildung 30: Innovationscontrolling-Menü) wird ein Menübaum mit der Bezeichnung „INNOVATIONSCONTROLLING“ für die Anwendung InnoCon erstellt. Diesem Menü werden die Masken (Funktionen) wie Projektidee, Folgedokumentation, Vorselektion und Projektidee bewerten zugeordnet.

Zuständigkeit: Basisbibliothek-Entwickler
NAV: Anwendungen Menüs

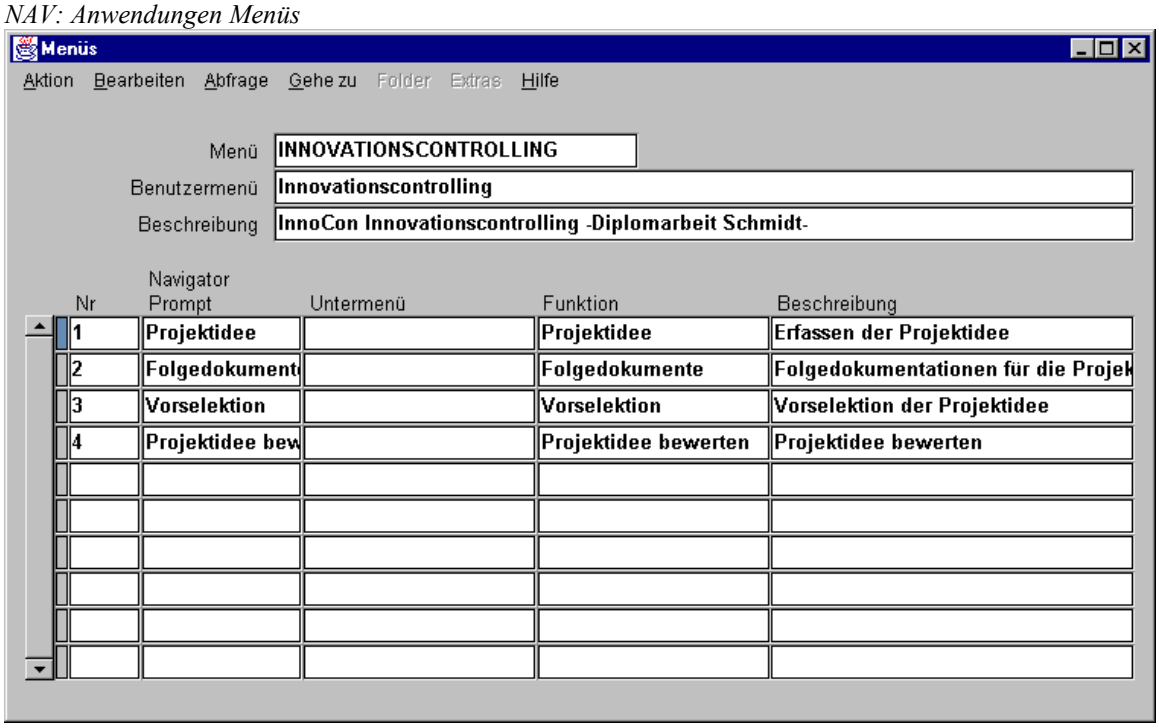


Abbildung 30: Innovationscontrolling-Menü

Um einen Menübaum für die Anwendung InnoCon zu erstellen, wird in der folgenden Abbildung ein weiteres Menü erstellt, das um die Untermenüs „INNOVATIONSCONTROLLING“ und „Hintergrundprozesse starten und anzeigen“ sowie die Funktionenen „Mitarbeiter bei caesar“, „Profiloptionen“ und Workflow-Benachrichitugnen“ erweitert wird.

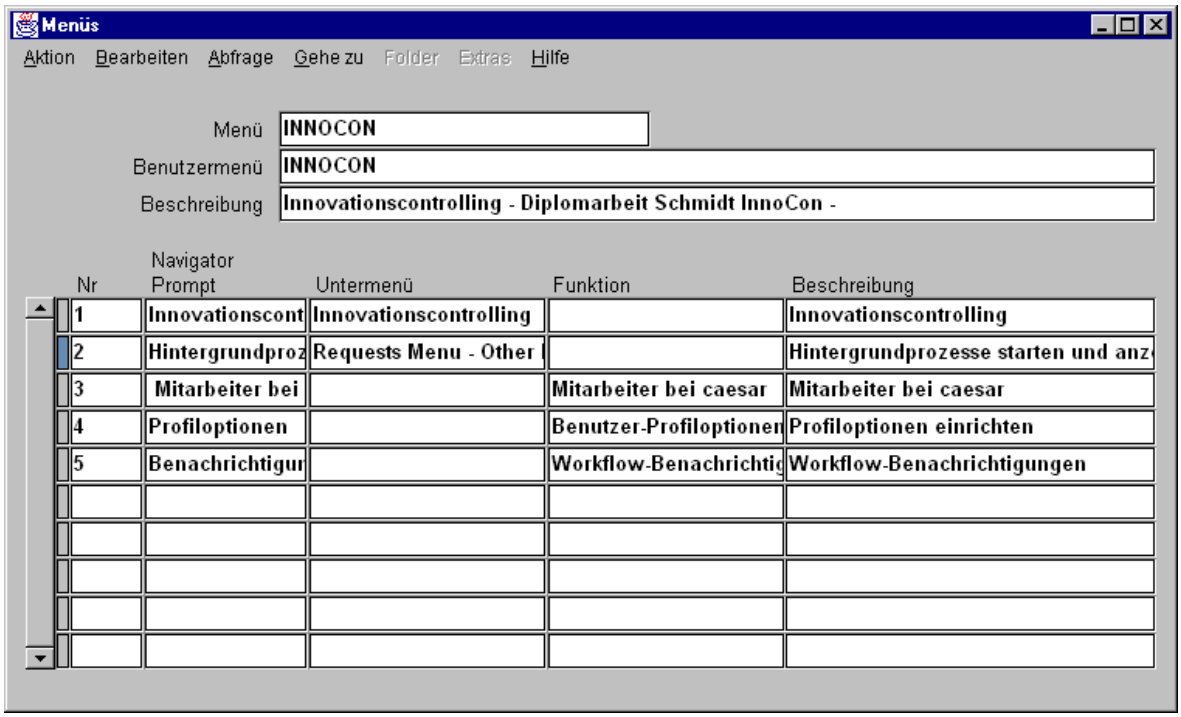


Abbildung 31: InnoCon-Menü

7.8 Zugriff auf InnoCon

Um die Anwendung InnoCon für die Mitarbeiter zugänglich zu machen wurde eine Zuständigkeit „InnoCon“ definiert. Diese Zuständigkeit muss den Mitarbeitern, die mit dieser Anwendung arbeiten werden, zugeordnet werden.

In der Maske Benutzer (vgl. Abbildung 32: Zugriff auf InnoCon) erfolgt der Eintrag welchem Benutzer von *Oracle Applications* der Zugriff auf die Anwendung (Zuständigkeit) InnoCon erteilt wird.

Zuständigkeit: *Basisbibliothek-Systemadministrator*
Navigation: *Sicherheitsregeln / Benutzer*

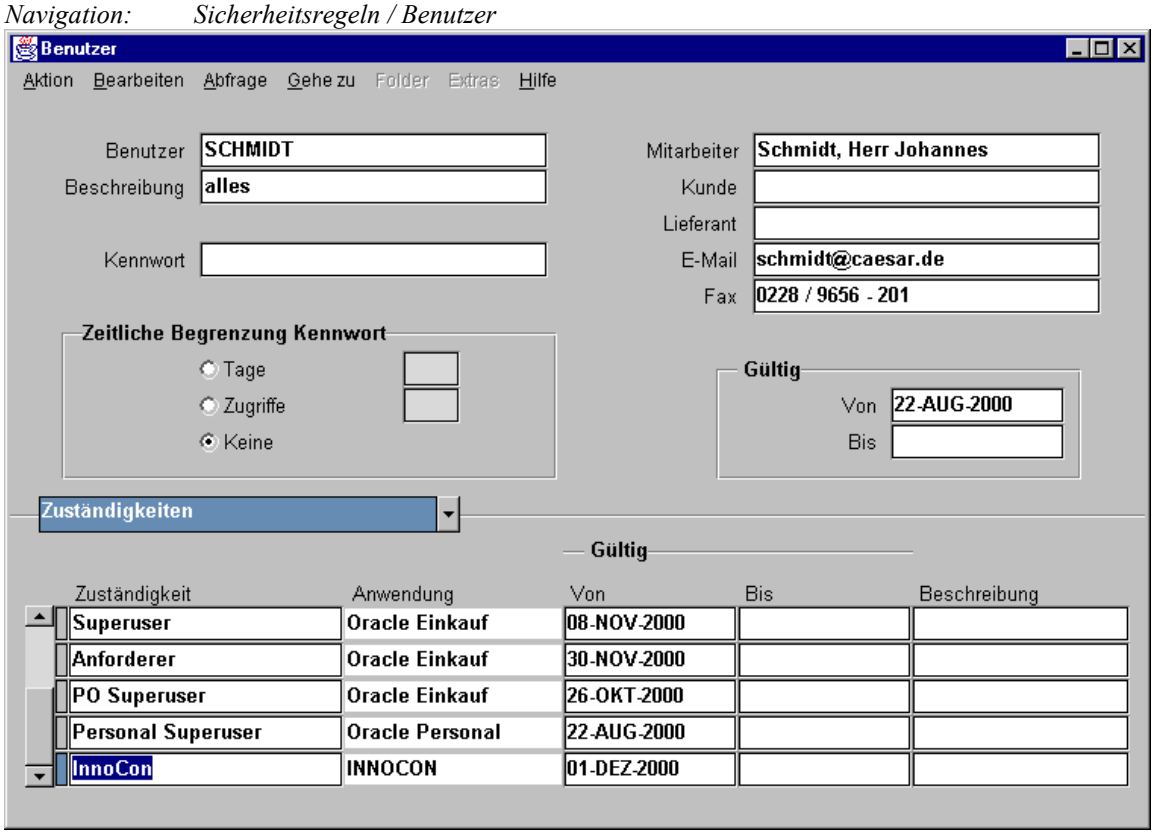


Abbildung 32: Abbildung: Zugriff auf InnoCon

8. Beschreibung der Anwendung InnoCon

Nachdem alle Setup-Schritte in *Oracle Applications* erfolgt sind, können Projektideen in dem Modul InnoCon erfasst werden. Nach der Auswahl der Zuständigkeit InnoCon erhält der Benutzer folgende Möglichkeiten:

1. Projektidee erfassen
2. Folgedokumente erfassen
3. Vorselektionsentscheidung über die Projektidee
4. Projektidee bewerten
5. Berichte ausdrucken

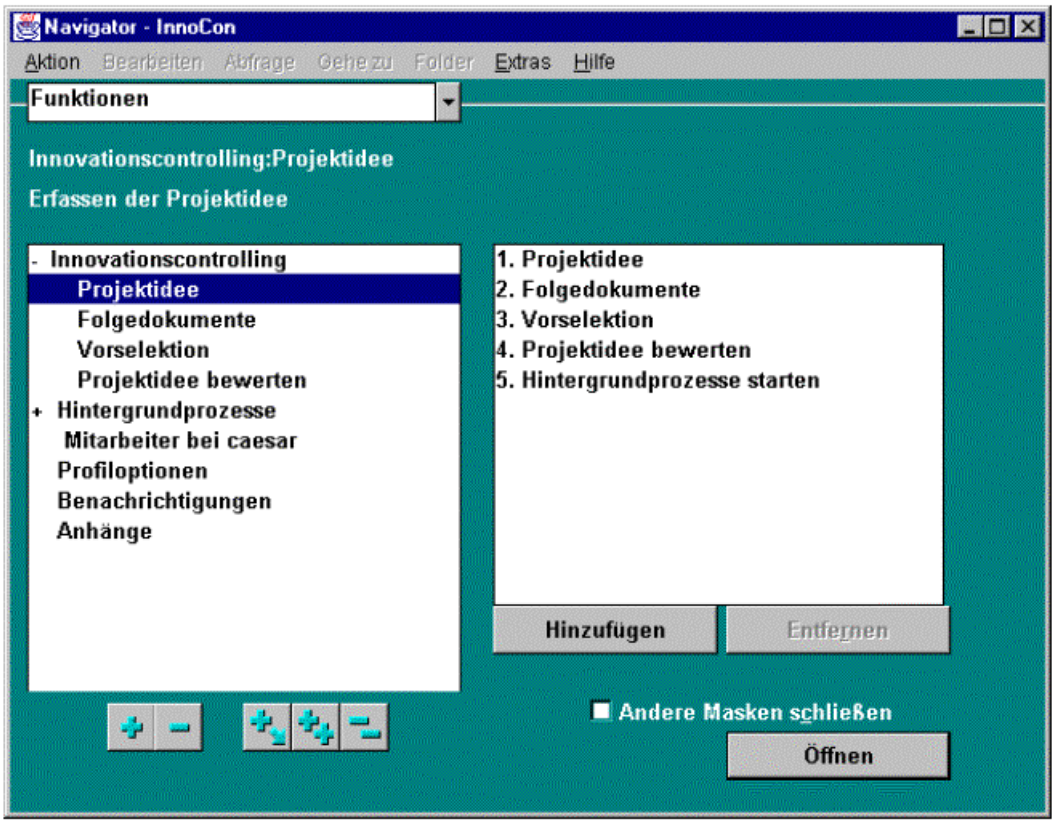


Abbildung 33: Navigator: InnoCon

8.1 Masken

8.1.1 Projektidee erfassen

In der folgenden Maske werden Projektideen erfasst. Nach dem Drücken der Schaltfläche „Neue Projektidee“ wird eine neue Projektnummer sowie das Datum der Ersterfassung (Systemdatum) automatisch angelegt.
Die Navigation zu den einzelnen Unterpunkten erfolgt über die TAB-Taste oder über die Schaltflächen rechts in der Maske.

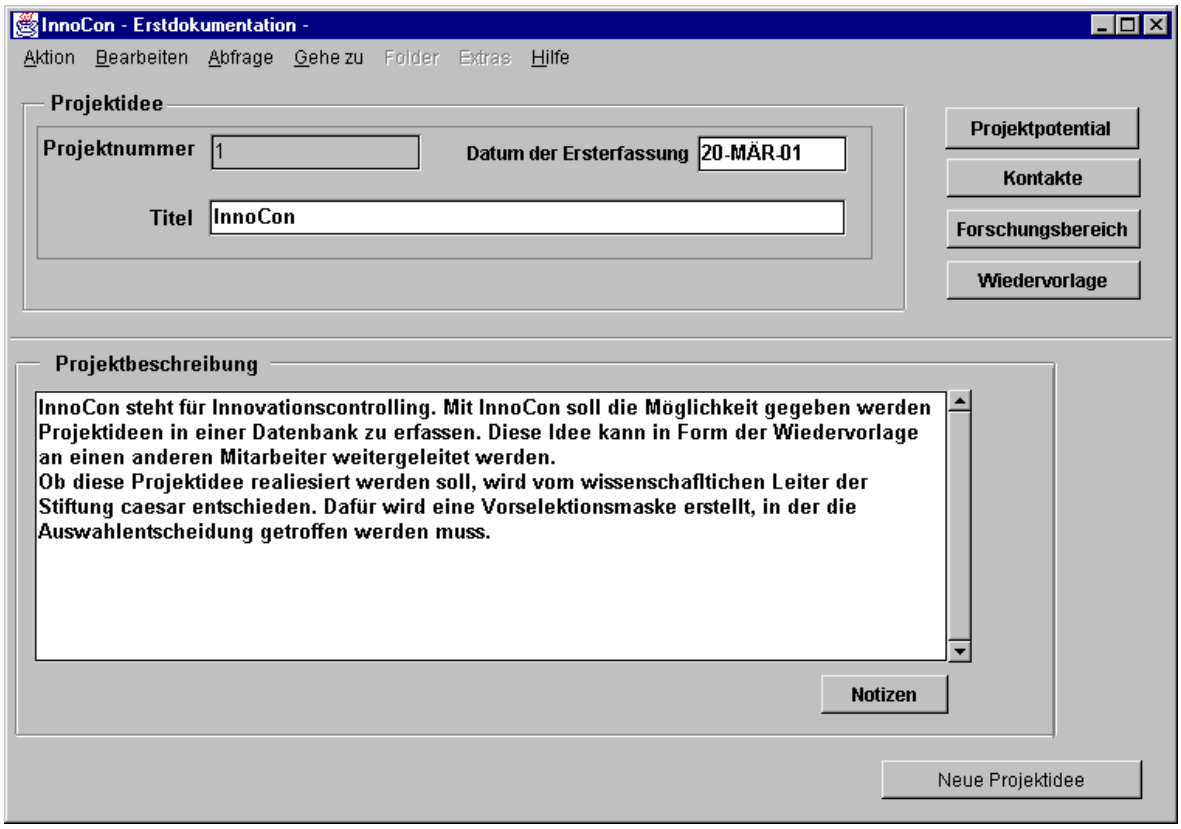


Abbildung 34: Maske: Erstdokumentation

Projektidee suchen

Um eine erfasste Projektidee zu suchen, wird in der Symbolleiste von *Oracle Applications* das Werteliste-Symbol gedrückt. Danach erscheinen alle erfassten Projektideen sortiert nach der Projektnummer. Hier kann die Auswahl der gesuchten Projektidee erfolgen um Änderungen durchführen.

8.1.2 Verknüpfungen

Um verschiedene Dateien beispielsweise Word- oder Excel-Dateien mit der Projektidee zu verknüpfen und in der Datenbank oder im Verzeichniss des ERP-Servers zu speichern, lässt sich die Funktion „Dokumentenanhänge“ durch Drücken der „Büroklammer“ in der Symbolleiste von *Oracle Applications* aktivieren. (Vgl. Kapitel „Oracle Applictions“ Abbildung 16: *Oracle Applications* - Symbolleiste).

Danach erscheint wie in der folgenden Abbildung dargestellt, die Maske für die Verknüpfung der Dokumente:

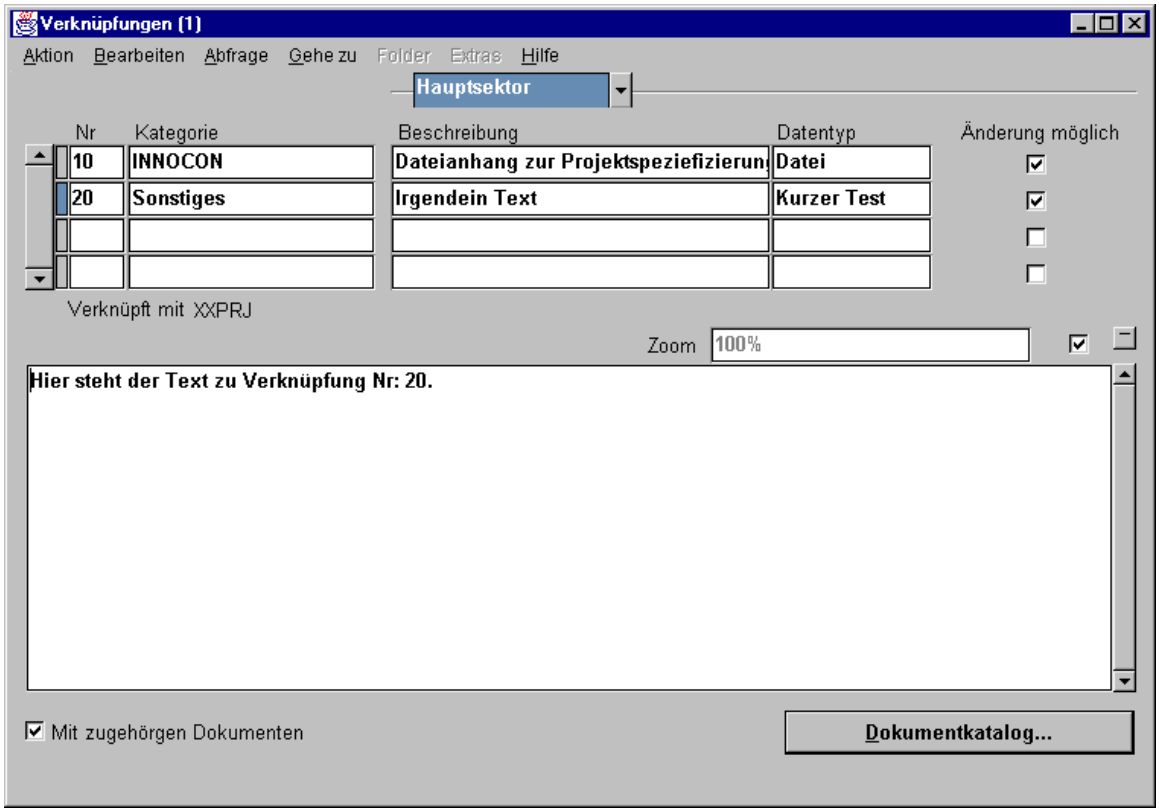


Abbildung 35: Verknüpfungen

Hier können die zugehörigen Dateien für die Projektidee im Verzeichnis des ERP-Servers gespeichert werden.

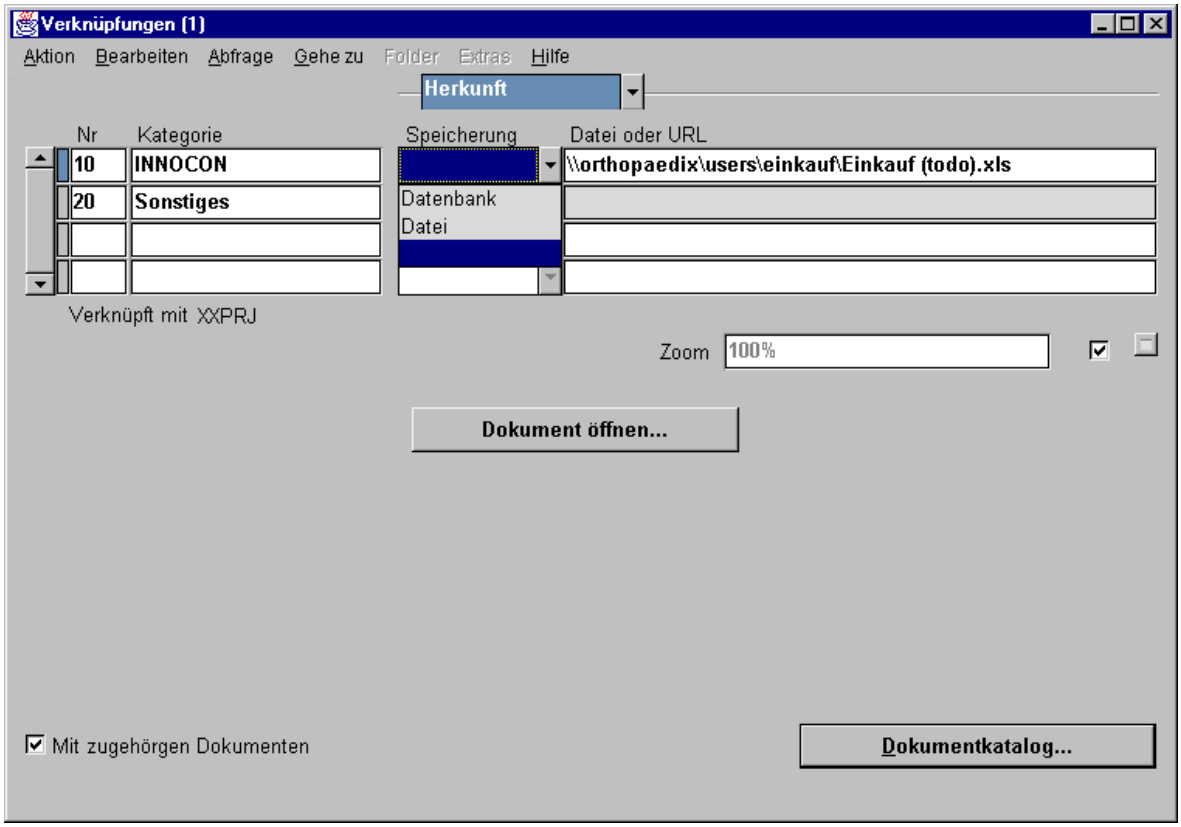


Abbildung 36: Verknüpfungen mit Datei oder URL

Will man die verknüpfte Datei öffnen, wird der Button „Dokument öffnen“ betätigt. Danach erscheint der Netscape Navigator, in dem die Bestätigung für das Öffnen des Dokumentes erfolgt. Erst dann öffnet sich das zugehörige Programm, je nach Dateityp Excel oder Word, hier hat man dann die Möglichkeit Änderungen durchzuführen oder das Dokument auszudrucken.

8.1.3 Folgedokumente

Die Erfassung der Folgedokumente geschieht in der folgenden Maske, die ähnlich aufgebaut ist, wie die vorherige (vgl.Abbildung 37: Maske Projektidee). Die Navigation zu den einzelnen Unterpunkten erfolgt jedoch über eine Auswahlliste im rechten Feld:

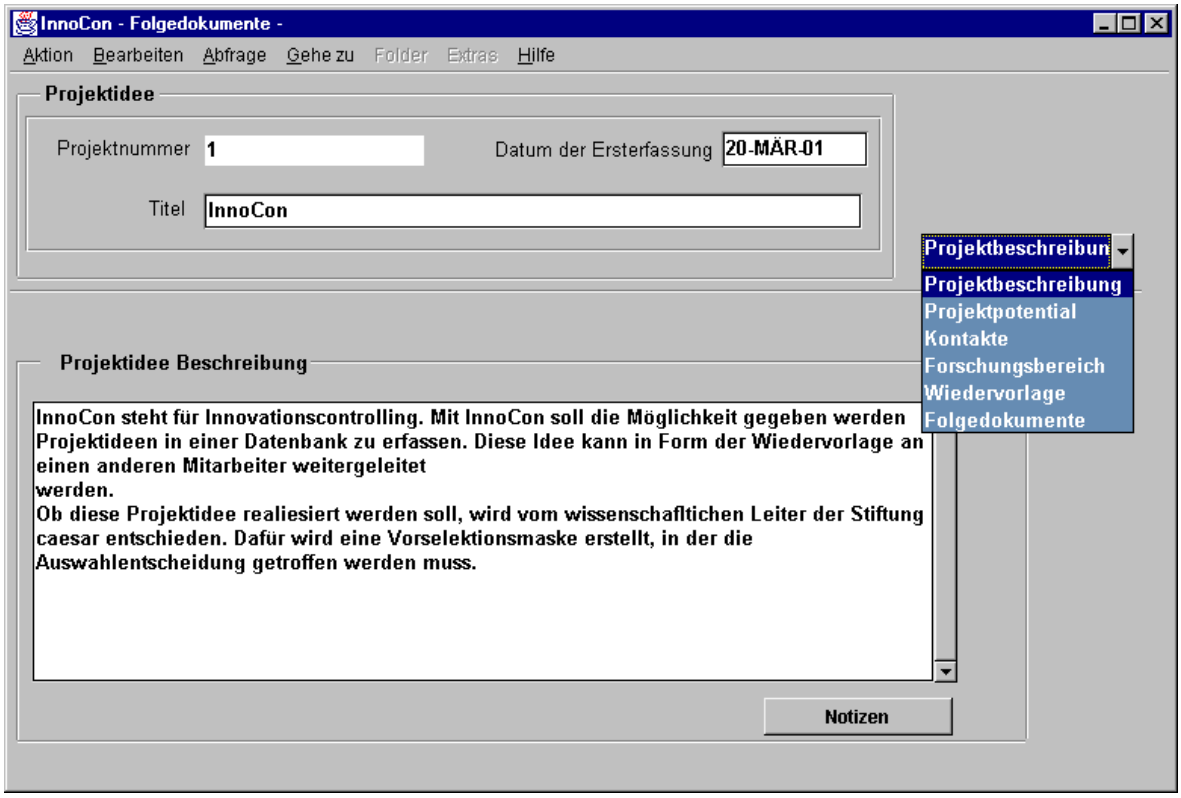


Abbildung 37: Folgedokumente

8.1.4 Vorselektion

In der Maske Vorselektion wird entschieden, ob die Idee weiterverfolgt wird:

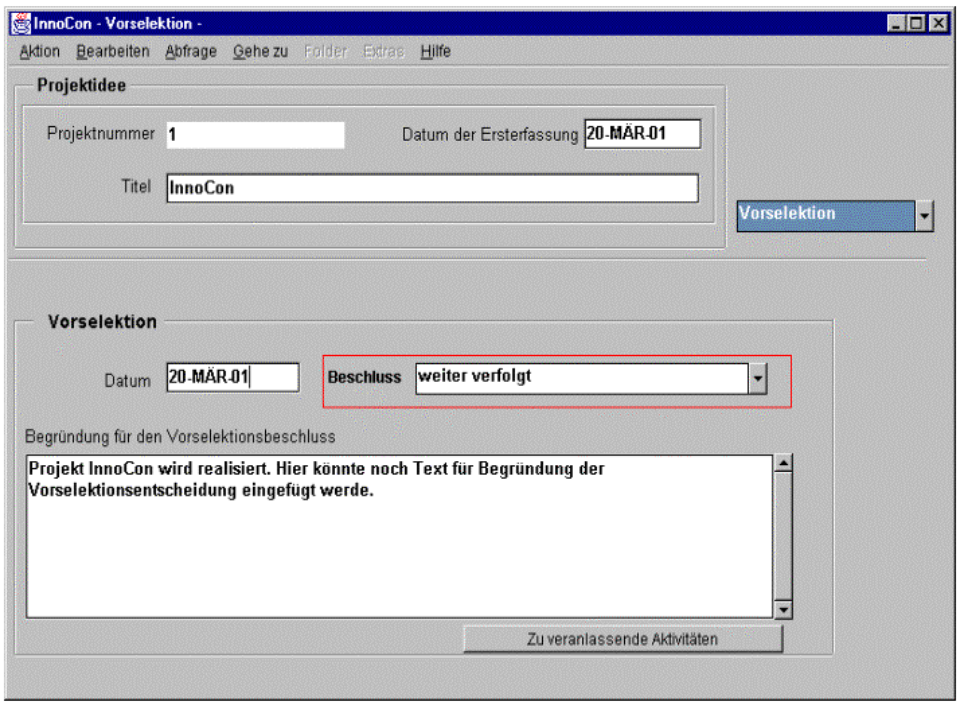


Abbildung 38: Maske: Vorselektion

Falls die Idee weiterverfolgt wird, kann diese nach dem Aufrufen der Maske „*Projektidee bewerten*“ (vgl. Abbildung 39: Bewertung der Projektideen) nach verschieden Kriterien bewertet werden.

8.1.5 Bewertung der Projektidee

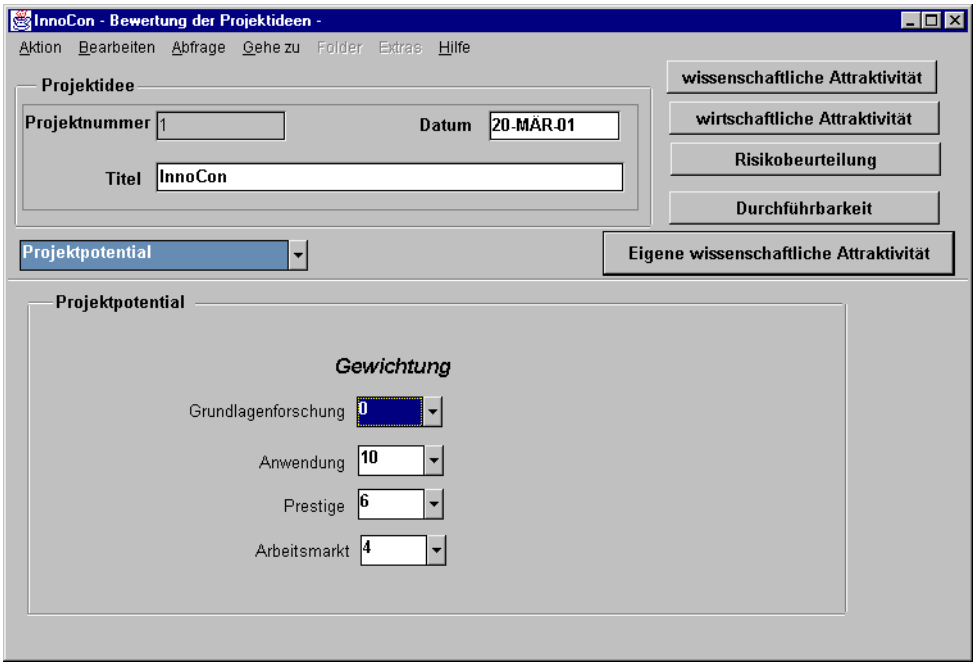


Abbildung 39: Bewertung der Projektideen

Alle Bewertungsmasken sind gleich aufgebaut. In der Gesamtbeurteilung hat man die Auswahl zwischen 1 und 15 zu treffen, während in den Gewichtungsfaktorspalten ein Wert eingesetzt wird, dessen Summe genau 100 ergeben muss. In der Spalte „Beitrag zur Gesamtattraktivität“ wird die Gesamtbeurteilung mit dem Gewichtungsfaktor multipliziert. Der Wert aus dem Beitrag zur Gesamtattraktivität dient zur Beurteilung der Projektideen.

Bewertung der wissenschaftlichen Attraktivität:

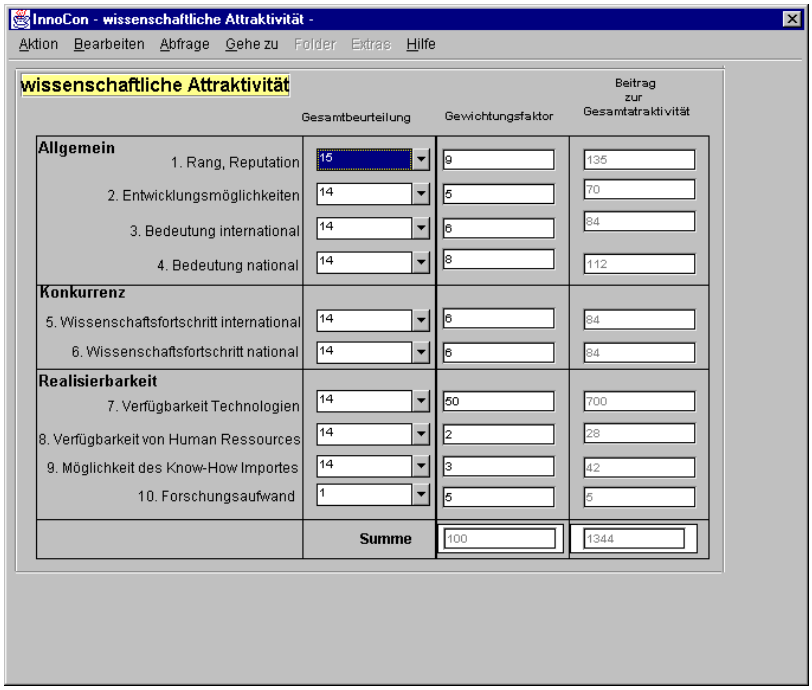


Abbildung 40: Maske: Bewertung der wissenschaftlichen Attraktivität

Bewertung der wirtschaftlichen Attraktivität:

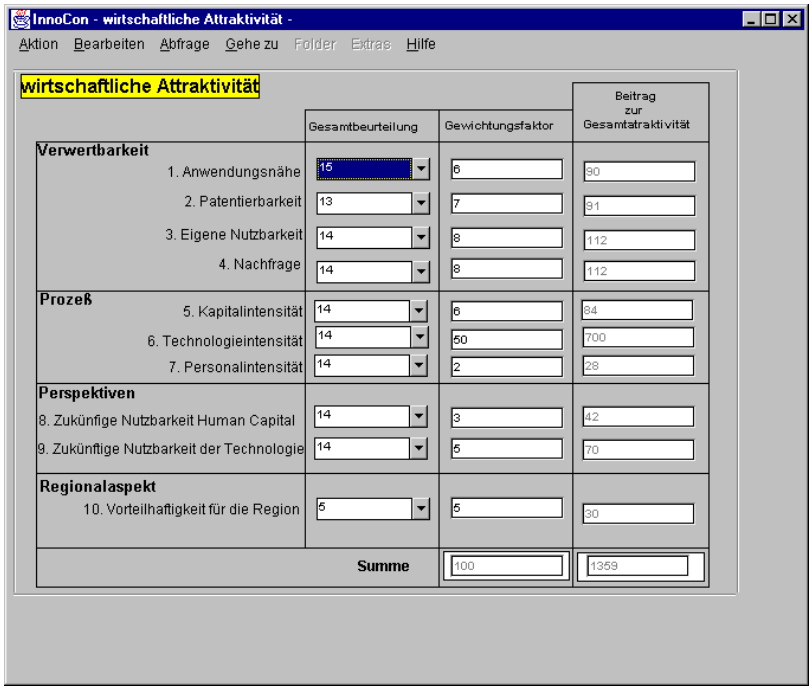


Abbildung 41: Maske: Bewertung der wirtschaftlichen Attraktivität

Risikobeurteilung kritischer Faktoren:

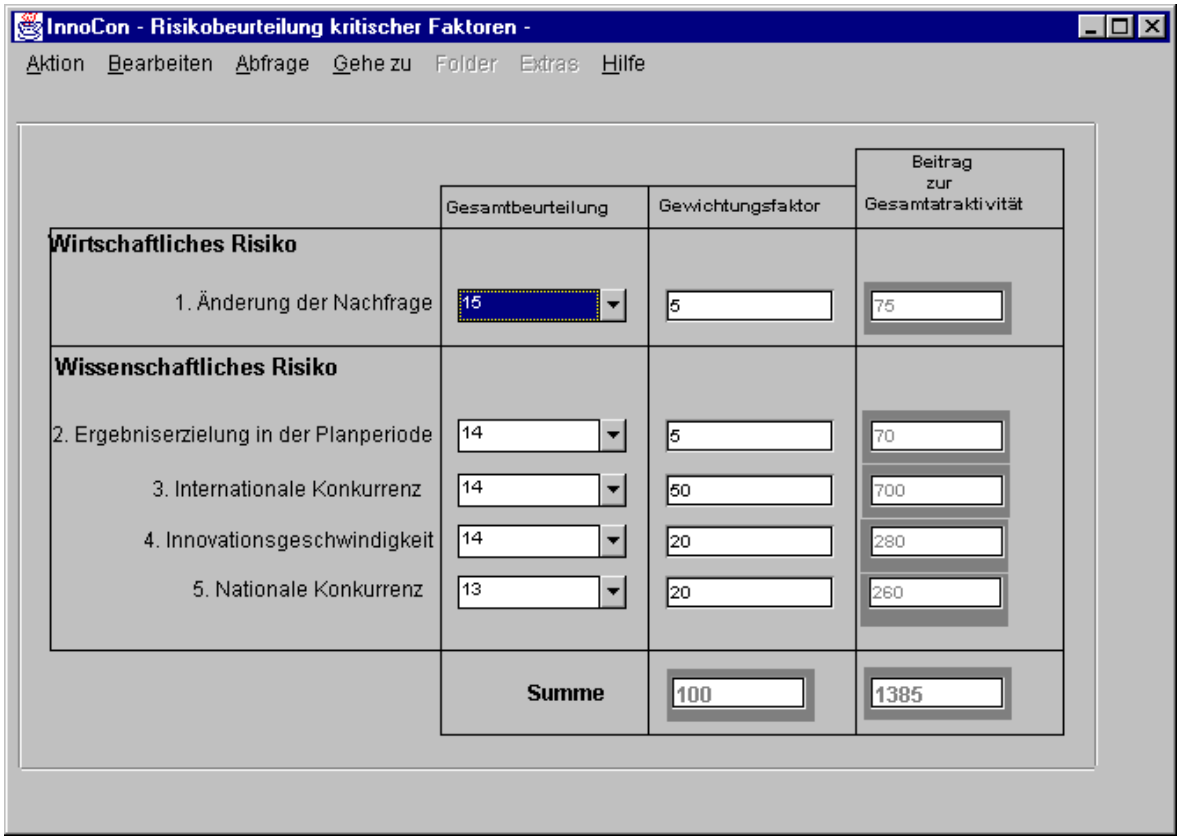


Abbildung 42: Maske: Risikobeurteilung kritischer Faktoren

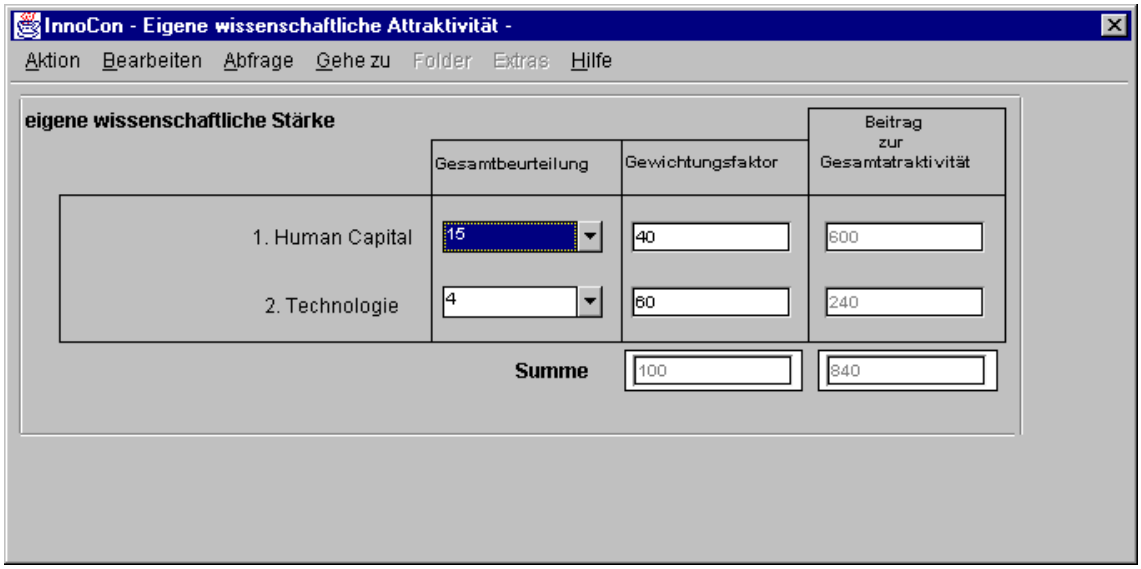


Abbildung 43: Maske: Bewertung der Eigenen wissenschaftlichen Attraktivität

8.2 Berichte

Um die erfassten Projektideen oder die Folgedokumente für bestimmte Projektidee auf dem Papier festzuhalten, müssen über Hintergrundprozesse die vorgegebenen Berichte gestartet werden. Die Ausgabe erfolgt als PDF-Dokument im Netscape Navigator. Der Vorteil dieser Ausgabeart begründet sich in der Möglichkeit, verschiedene Schriftarten und Zeichen in Reports abzubilden.
Über den Netscape Navigator hat man die Möglichkeit, die Berichte auszudrucken.

8.2.1 Report Erstdokumentation

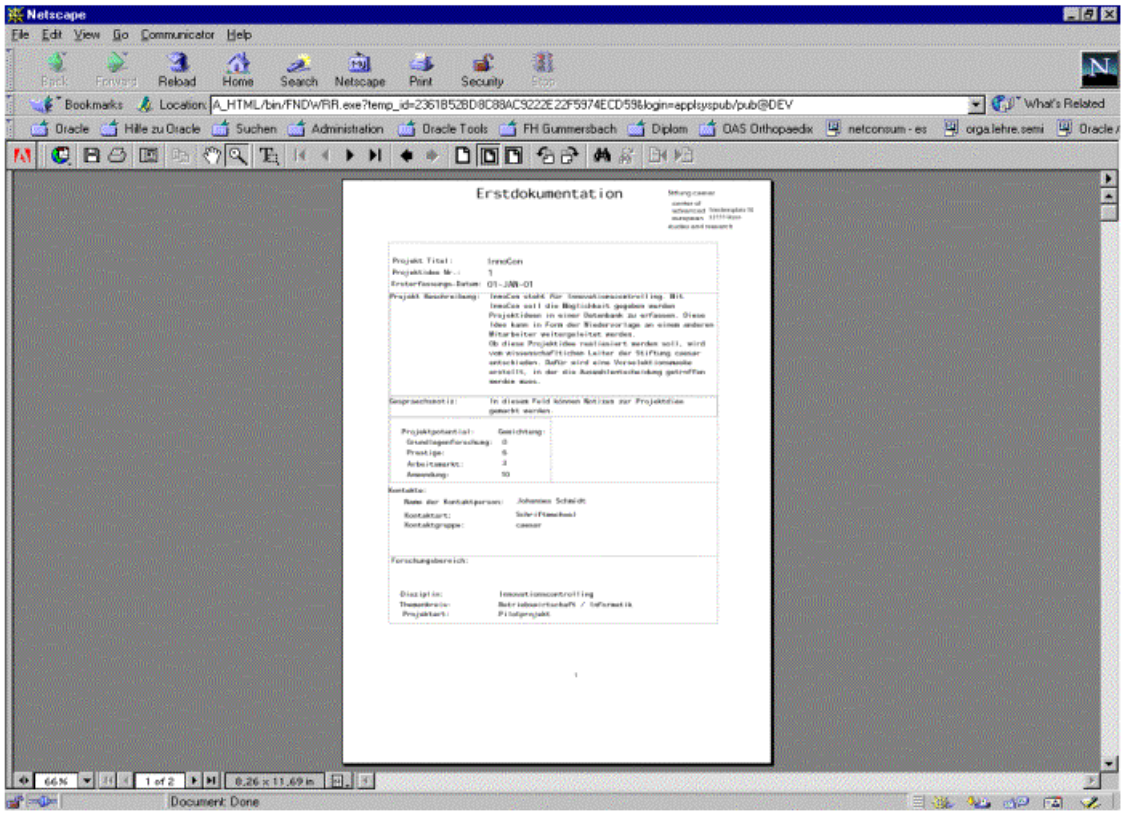


Abbildung 44: Report: Erstdokumentation

8.2.2 Bericht Folgedokumente

Die Ausgabe der Folgedokumente erfolgt im folgenden Report (vgl. Abbildung 45: Report Folgedokumente). Alle Folgedokumente zu dieser Projektidee werden hier untereinander sortiert nach Datum angezeigt.

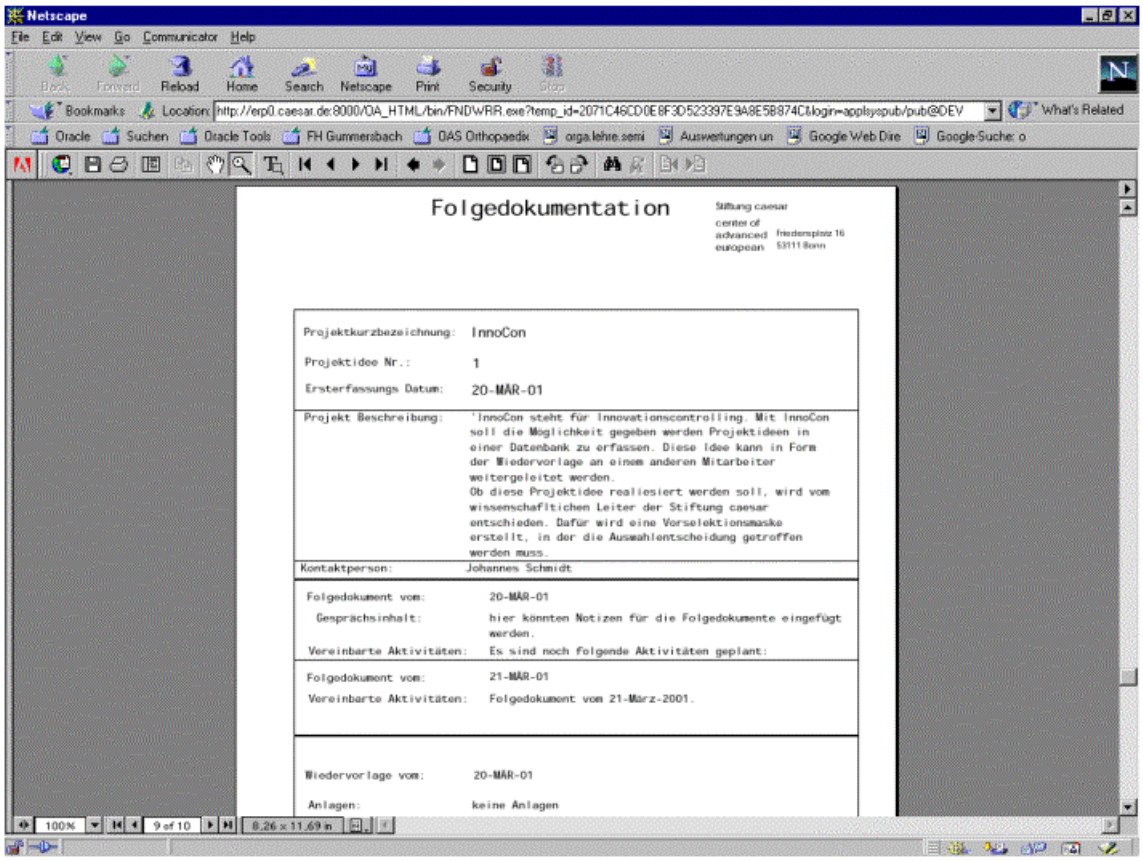


Abbildung 45: Report: Folgedokumente

8.2.3 Bericht Vorselektion

Die Ausgabe der Vorselektion erfolgt im folgenden Report (vgl. Abbildung 45: Report Vorselektion). Alle Vorselektionsentscheidungen zu dieser Projektidee werden hier untereinander sortiert nach Datum angezeigt:

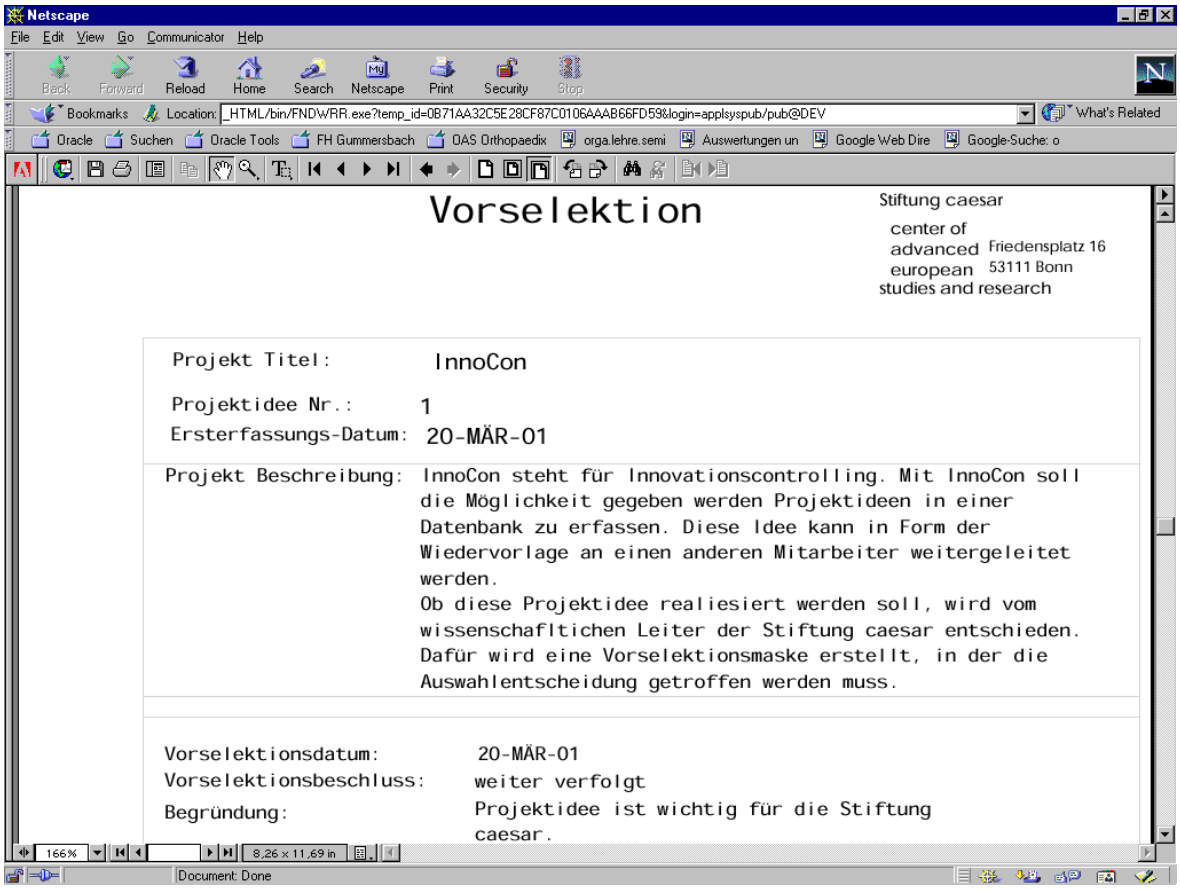


Abbildung 46: Report: Vorselektion

8.2.4 Bericht Projektvorlauf

In diesem Bericht wird der Verlauf von der Erstdokumentation bis zur Fertigstellung für das einzelne Projekt dokumentiert (vgl. Abbildung 47: Report Projektvorlauf):

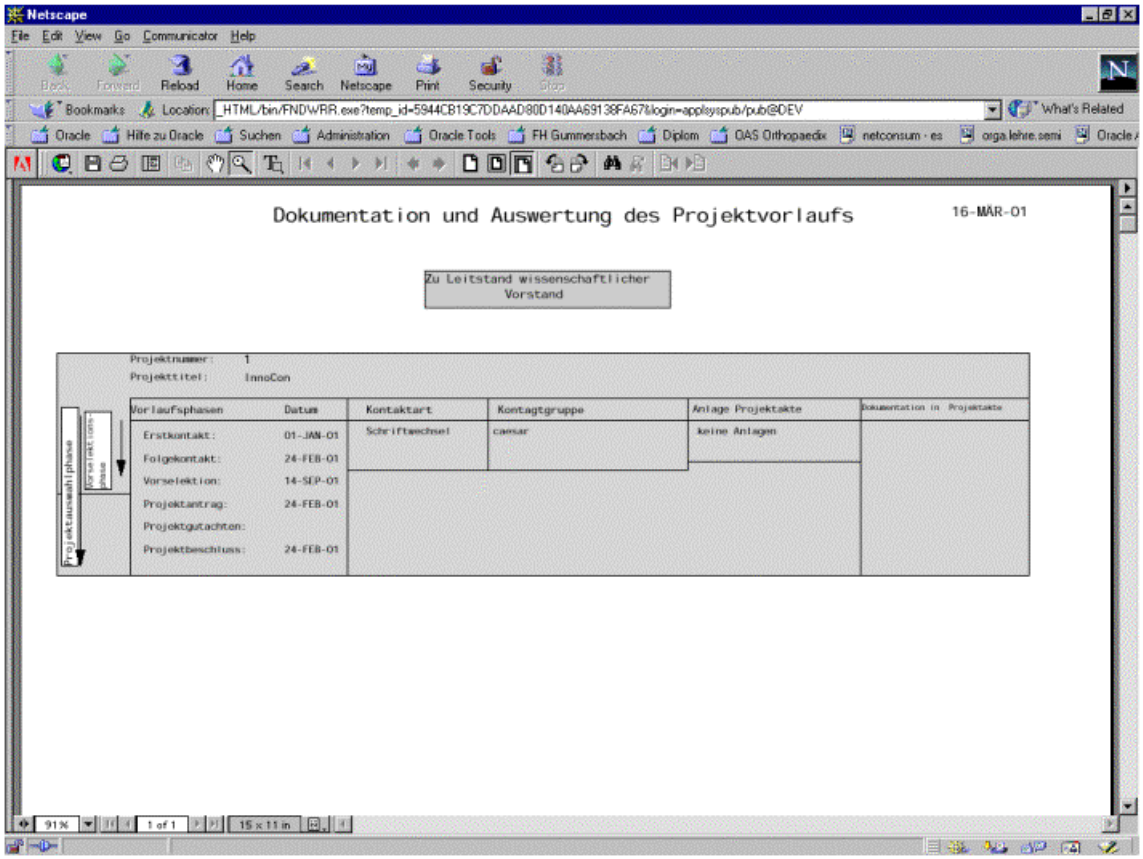
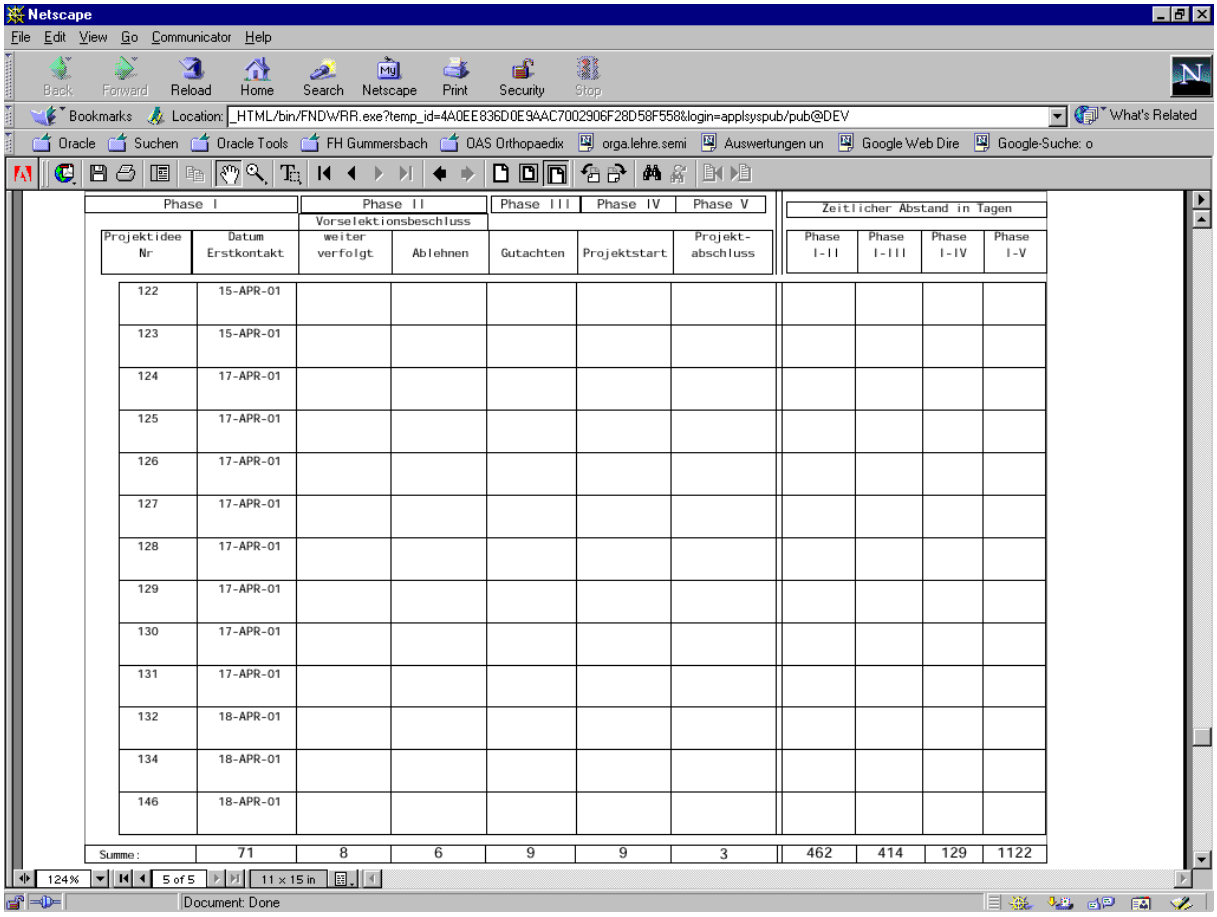


Abbildung 47: Report: Projektvorlauf

8.2.5 Projektverlaufzeitanalyse

In der Projektverlaufzeitanalyse wird die Innovationsgeschwindigkeit durch die Darstellung des zeitlichen Abstandes von der Ersterfassung bis zur Realisierung der Projekte dargestellt (vgl.Abbildung 48:Projektverlaufzeitanalyse). Die Projektphasen (Phase I bis Phase V) zeigen den Projektstatus an.



Phase I		Phase II		Phase III	Phase IV	Phase V	Zeitlicher Abstand in Tagen			
		Vorselektionsbeschluss								
Projektidee Nr	Datum Erstkontakt	weiter verfolgt	Ablehnen	Gutachten	Projektstart	Projekt-abschluss	Phase I-II	Phase I-III	Phase I-IV	Phase I-V
122	15-APR-01									
123	15-APR-01									
124	17-APR-01									
125	17-APR-01									
126	17-APR-01									
127	17-APR-01									
128	17-APR-01									
129	17-APR-01									
130	17-APR-01									
131	17-APR-01									
132	18-APR-01									
134	18-APR-01									
146	18-APR-01									
Summe:	71	8	6	9	9	3	462	414	129	1122

Abbildung 48: Report: Projektverlaufzeitanalyse

Teil C: Anhang

9. Zusammenfassung und Ausblick

Ziel dieser Diplomarbeit war es, ein Innovationscontrolling-Instrument zu erstellen, in dem Projektideen der Stiftung caesar erfasst und bewertet werden können.

Dies hätte man auch mit einer *Oracle Forms* Anwendung realisieren können. Hierfür müssten jedoch zusätzliche Lizenzen für eine Oracle Datenbank, Oracle-Forms und Oracle-Reports sowie dem Web Application Server von Oracle erworben werden.
Die Anwendung InnoCon wurde aus folgenden Gründen in *Oracle Applications* implementiert:
In der Stiftung caesar wird *Oracle Applications* bereits eingesetzt, dadurch müssen Daten, beispielsweise Mitarbeiter der Stiftung caesar, nicht doppelt erfasst werden. Auch die Wartungsarbeiten für die Datenbankanwendung werden gesichert, da ein Systemverwalter für Oracle Applications zuständig ist.
Auch die Schnittstellen zu anderen Modulen, wie beispielsweise Projektfaktura, ist zukünftig leichter zu implementieren. Ausserdem hat der Autor der vorliegenden Diplomarbeit Erfahrungen mit *Oracle Applications* sammeln können.

Mit InnoCon ist der erste Baustein für das Innovationscontrolling der Stiftung caesar realisiert worden. Da die Projektideenerfassung mittels einer Datenbankanwendung in *Oracle Applications* ermöglicht wurde.
InnoCon liefert die Daten, die die Basis für die dann folgenden Module (Projektfaktura) des Controllingkonzeptes der Stiftung caesar darstellen.

Jedoch wird die Erweiterbarkeit dieses Projektes erwartet. Es sollte noch die Projektidee-Erfassung als Benachrichtigung an andere Mitarbeiter der Stiftung caesar weitergeleitet werden können. Hierzu sollte die Anwendung InnoCon in das Workflow-System von *Oracle Applications* eingebunden werden.

Zusätzlich sollte eine Schnittstelle zu *Oracle Applications* Projektfaktura erstellt werden. Mit dieser Schnittstelle sollen Projektideen, die in InnoCon bewertet und realisiert werden sollen, in das Modul Projektfaktura importiert werden.

Probleme

Die Einbindung der Funktion „Attachments“ (Dokumentenanhänge) in die Anwendung InnoCon, in der Dokumente auf dem ERP-Server oder in der Datenbank zu speichern bereitete insofern Schwierigkeiten, als dass es zu diesem Setup-Schritt keine Dokumentation zur Verfügung stand. Auch für die Einbindung der Graphischen Objekte, beispielsweise das Logo von caesar, existieren keine Dokumentationen. Grundsätzlich ist Oracle sehr sparsam mit Informationen zu der Anwendung *Oracle Applications*.

Oracle Support konnte für die Implementierung der Anwendung InnoCon nicht hinzugezogen werden, da die Entwicklung eigener Anwendungen für die Stiftung caesar nicht zum Support Umfang gehört.

9.1 Abbildungsverzeichnis

9.2 Tabellenverzeichnis

9.3 Literatur- und Quellenverzeichnis

[Baum, Coenenberg, Günther, 1999]	H.-G. Baum, A. G. Coenenberg, T. Günther „Strategisches Controlling“ Schäffer Poeschel Verlag
[Boutellier, Völker, Voit, 1999]	Roman Boutellier, Rainer Völker, Eugen Voit: „Innovationscontrolling“ Hanser Verlag
[Faeskorn-Woyke, 2000]	Prof. Dr. Heide Faeskorn-Woyke "Vorlesungsscript zum Fach Datenbanken" Fachhochschule Köln – Abtlg. Gummersbach, 2000
[Hermann, Lenz, Unbesch 1999]	Uwe Herrmann, Dierk Lenz, Günter Unbescheid “Oracle8 für den DBA“ Addisson-Wesley Verlag
[Hoppe-Mempel, 1998]	Ralph Hoppe, Matthias Memepel "Oracle Designer R2.1" Addisson-Wesley Verlag
[Hoffmann, 2000]	Prof. Dr. Karl-Heinz Hoffmann "Mission von caesar" Stiftung caesar
[Hugenroth, 2000]	Francis Hugenroth „Forschen in neuen Dimensionen“ Stiftung caesar
[Lockman, 1998]	David Lockman “Oracle8 Datenbankentwicklung in 21 Tagen” Markt+Technik Verlag
[Mueller, 1998]	Robert J. Muller "Oracle Developer/2000 Handbook" Oracle Press
[Oracle, 1998]	*Oracle Applications* “Develper’s Guide Release 11” Oracle Press
[Ponndorf, Matthäus, 2001]	Steven Ponndorf, Wolf-Gert Matthäus

	„Oracle Devloper 6i“ Addison-Wesley
[Scheer 1991]	A. W. Scheer „Wirtschaftsinformatik: Informationssystem im Industriebetrieb“ Springer-Verlag
[Stippel, 1999]	Nocola Stippel „Innovationscontrolling“ Verlag Vahlen
[Steinle, Bruch 1998]	C. Steinle, H. Bruch „Controlling“ Schäffer Poeschl Verlag, Stuttgart
[Urman, 2000]	Scott Urman „Oracle PL/SQL Programmierung“ Hanser Verlag
[Wagner, 1999]	Thomas Wagner „Wissenschaftliche Projekte“ Stiftung caesar
[Wöhe, 1986],	Günter Wöhe "Einführung in die allgemeine Betriebswirtschaftslehre" 16. Auflage, Vahlen Verlag, München
[Ziegenbein, 1998]	Prof. Dr. Klaus Ziegenbein "Controlling" Kiehl
Internetquellen:	
[Gabler-Online]	http://www.gabler-online.de/lexikon-controlling
[Oracle]	http://www.oracle.de
[*Oracle Applications*, 2001]	*Oracle Applications* Online Documentation

9.4 Inhalt der CD-ROM

Die CD-ROM, die dieser Diplomarbeit beiliegt, enthält folgende Verzeichnisse :

Verzeichnis „Diplomarbeit“

- Diplomarbeit.doc
- Diplomarbeit.pdf

Verzeichnis „Innocon“ enthält den Quellcode für die Anwendung InnoCon, untergliedert in folgende Verzeichnisse:

- Reports
- lib
- Forms
- Syntax

Verzeichnis „Präsentation“

- InnoCon Präsentation.ppt

9.5 Quellcode

Tabellenbeschreibung

XXPRJ = Projektidee
XXKP = Kontaktperson
XXKONT = Kontakte
XXFB = Forschungsbereich
XXPRJDOKU = Projektdokumente
XXVORS = Vorselektion
XXFOL = Folgedokumente
XXWV = Wiedervorlage

Tabellen erstellen

```
PROMPT Creating Table 'CG_REF_CODES'
CREATE TABLE CG_REF_CODES
 (RV_DOMAIN VARCHAR2(100) NOT NULL
 ,RV_LOW_VALUE VARCHAR2(240) NOT NULL
 ,RV_HIGH_VALUE VARCHAR2(240)
 ,RV_ABBREVIATION VARCHAR2(240)
 ,RV_MEANING VARCHAR2(240)   )
/

PROMPT Creating Table 'XXPRJ'
CREATE TABLE XXPRJ
 (PROJEKT_NUMMER NUMBER(10) NOT NULL
 , TITEL VARCHAR2(100)
 , DATUM DATE DEFAULT SYSDATE NOT NULL
 , BESCHREIBUNG VARCHAR2(1000)
 , GESPRAECHSNOTIZ_ERSTDOK VARCHAR2(1000)
 , CREATED_BY NUMBER(15)
 , CREATION_DATE DATE DEFAULT SYSDATE NOT NULL
 , LAST_UPDATED_BY NUMBER(15)
 , LAST_UPDATED_DATE DATE DEFAULT SYSDATE NOT NULL
 , LAST_UPDATED_LOGIN NUMBER(15)
 , XXKONT_XXKONT_ID NUMBER(10, 0)
 , XXFB_XXFB_ID NUMBER(10, 0)
, XXPRJ_ID NUMBER(10, 0) NOT NULL )
/

COMMENT ON COLUMN XXPRJ.PROJEKT_NUMMER IS 'Projekt Nummer / Primärschlussel'
/
COMMENT ON COLUMN XXPRJ.TITEL IS 'Kurzer Name oder Kurzbezeichnung für das Projekt'
/
COMMENT ON COLUMN XXPRJ.DATUM IS 'Ersterfassungsdatum'
/
COMMENT ON COLUMN XXPRJ.BESCHREIBUNG IS 'Kurzbeschreibung'
/

PROMPT Creating Table 'XXKP'
CREATE TABLE XXKP
 (ANREDE VARCHAR2(100)
 , TITEL VARCHAR2(100)
 , NAME VARCHAR2(100)
 , GESCHLECHT VARCHAR2(20)
 , CREATED_BY NUMBER(15)
 , CREATION_DATE DATE DEFAULT SYSDATE NOT NULL
```

```
 , LAST_UPDATED_BY NUMBER(15)
 , LAST_UPDATED_DATE DATE DEFAULT SYSDATE NOT NULL
 , LAST_UPDATED_LOGIN NUMBER(15)
 , XXKPADR_XXKPADR_ID NUMBER(10, 0)
 , XXKP_ID NUMBER(10, 0) NOT NULL )
/

PROMPT Creating Table 'XXPRJDOKU'
CREATE TABLE XXPRJDOKU
 (PLAN_BUDGET NUMBER(10)
 , PLAN_PERSONAL VARCHAR2(20)
 , PRJ_ANTRAG_DATUM DATE
 , PRJ_ANTRAG_RUECK_DATUM DATE
 , PRJ_ANTRAG_DATEIANHANG VARCHAR2(70)
 , BESCHLUS_DATUM DATE
 , BESCHLUS_KURZTEXT VARCHAR2(200)
 , GUTACHTEN_DATUM DATE
 , GUTACHTEN_BESCHREIBUNG VARCHAR2(200)
 , CREATED_BY NUMBER(15)
 , CREATION_DATE DATE DEFAULT SYSDATE NOT NULL
 , LAST_UPDATED_BY NUMBER(15)
 , LAST_UPDATED_DATE DATE DEFAULT SYSDATE NOT NULL
 , LAST_UPDATED_LOGIN NUMBER(15)
 , XXPRJ_XXPRJ_ID NUMBER(10, 0) NOT NULL
 , XXPRJDOKU_ID NUMBER(10, 0) NOT NULL )
/
COMMENT ON COLUMN XXPRJDOKU.PLAN_BUDGET IS 'Erfassen des nötigen Kapitals'
/

COMMENT ON COLUMN XXPRJDOKU.PLAN_PERSONAL IS 'Wird Personal benötigt,  oder hat caesar
die nötigen Mitarbeiter?'
/
COMMENT ON COLUMN XXPRJDOKU.PRJ_ANTRAG_RUECK_DATUM IS 'Datum,  wann der
Projektantrag zurückkam'
/


PROMPT Creating Table 'XXKONT'
CREATE TABLE XXKONT
 (KONTAKTGRUPPE VARCHAR2(240)
 , SONSTIGE_KGR VARCHAR2(240)
 , KONTAKTART VARCHAR2(240)
 , SONSTIGE_KA VARCHAR2(240)
 , CREATED_BY NUMBER(15)
 , CREATION_DATE DATE DEFAULT SYSDATE NOT NULL
 , LAST_UPDATED_BY NUMBER(15)
 , LAST_UPDATED_DATE DATE DEFAULT SYSDATE NOT NULL
 , LAST_UPDATED_LOGIN NUMBER(15)
 , XXKONT_ID NUMBER(10, 0) NOT NULL )
/

COMMENT ON COLUMN XXKONT.KONTAKTGRUPPE IS 'Auswahl der  Kontaktgruppe'
/

COMMENT ON COLUMN XXKONT.KONTAKTART IS 'Auswahl der Kontaktart'
/

PROMPT Creating Table 'XXVORS'
CREATE TABLE XXVORS
 (DATUM DATE DEFAULT SYSDATE NOT NULL
```

```
, BEGRUENDUNG VARCHAR2(400)
, VORSELEKTIONSBESCHLUSS VARCHAR2(240)
, CREATED_BY NUMBER(15)
, CREATION_DATE DATE DEFAULT SYSDATE NOT NULL
, LAST_UPDATED_BY NUMBER(15)
, LAST_UPDATED_DATE DATE DEFAULT SYSDATE NOT NULL
, LAST_UPDATED_LOGIN NUMBER(15)
, XXPRJ_XXPRJ_ID NUMBER(10, 0) NOT NULL
, XXVORS_ID NUMBER(10, 0) NOT NULL )
/

COMMENT ON COLUMN XXVORS.DATUM IS 'Datum der Vorselektion'
/

COMMENT ON COLUMN XXVORS.BEGRUENDUNG IS 'Begründung,  was mit dem Projekt geschieht.'
/

COMMENT ON COLUMN XXVORS.VORSELEKTIONSBESCHLUSS IS 'Auswahl,  ob die Prj-Idee
weiterverfolgt wird?'
/

PROMPT Creating Table 'XXFB'
CREATE TABLE XXFB
 (TRIPLET VARCHAR2(240)
 , DISZIPLIN VARCHAR2(240)
 , THEMENKREIS VARCHAR2(240)
 , BEREICH VARCHAR2(240)
 , PROJEKTART VARCHAR2(240)
 , CREATED_BY NUMBER(15)
 , CREATION_DATE DATE DEFAULT SYSDATE
 , LAST_UPDATED_BY NUMBER(15)
 , LAST_UPDATED_DATE DATE DEFAULT SYSDATE
 , LAST_UPDATED_LOGIN NUMBER(15)
 , XXFB_ID NUMBER(10, 0) NOT NULL )
/

COMMENT ON COLUMN XXFB.BEREICH IS 'Auswahl zwischen 1, 2, 3'
/

PROMPT Creating Table 'XXKPADR'
CREATE TABLE XXKPADR
 (STRASSE VARCHAR2(100)
 , PLZ NUMBER(10)
 , ORT VARCHAR2(100)
 , TELEFON VARCHAR2(20)
 , EMAIL VARCHAR2(100)
 , XXKPADR_ID NUMBER(10, 0) NOT NULL )
/

PROMPT Creating Table 'XXWV'
CREATE TABLE XXWV
 (DATUM DATE DEFAULT SYSDATE NOT NULL
 , VERTEILER_AN VARCHAR2(100)
 , ANLAGEN VARCHAR2(100)
 , BEARBEITER VARCHAR2(100)
 , VERSAND_PRJANTRAG VARCHAR2(20)
 , CREATED_BY NUMBER(15)
 , CREATION_DATE DATE DEFAULT SYSDATE NOT NULL
 , LAST_UPDATED_BY NUMBER(15)
 , LAST_UPDATED_DATE DATE DEFAULT SYSDATE NOT NULL
```

```
, LAST_UPDATED_LOGIN NUMBER(15)
, XXVORS_XXVORS_ID NUMBER(10, 0)
, XXFOL_XXFOL_ID NUMBER(10, 0)
, XXPRJ_XXPRJ_ID NUMBER(10, 0)
, XXWV_ID NUMBER(10, 0) NOT NULL )
/

PROMPT Creating Table 'XXFOL'
CREATE TABLE XXFOL
 (DATUM DATE DEFAULT SYSDATE NOT NULL
 , GESPRAECHSNOTIZ VARCHAR2(400)
 , VEREINBARTE_AKTIVITAETEN VARCHAR2(400)
 , CREATED_BY NUMBER(15)
 , CREATION_DATE DATE DEFAULT SYSDATE NOT NULL
 , LAST_UPDATED_BY NUMBER(15)
 , LAST_UPDATED_DATE DATE DEFAULT SYSDATE NOT NULL
 , LAST_UPDATED_LOGIN NUMBER(15)
 , XXPRJ_XXPRJ_ID NUMBER(10, 0) NOT NULL
 , XXFOL_ID NUMBER(10, 0) NOT NULL )
```

Indexe

```
PROMPT Creating Index 'X_CG_REF_CODES_1'
CREATE INDEX X_CG_REF_CODES_1 ON CG_REF_CODES
 (RV_DOMAIN
 , RV_LOW_VALUE)
/

PROMPT Creating Index 'XXPRJ_XXKONT_FK_I'
CREATE INDEX XXPRJ_XXKONT_FK_I ON XXPRJ
 (XXKONT_XXKONT_ID)
/
PROMPT Creating Index 'XXPRJ_XXFB_FK_I'
CREATE INDEX XXPRJ_XXFB_FK_I ON XXPRJ
 (XXFB_XXFB_ID)
/
PROMPT Creating Index 'XXPRJ_XXKP_FK_I'
CREATE INDEX XXPRJ_XXKP_FK_I ON XXPRJ
 (XXKP_XXKP_ID)
/
PROMPT Creating Index 'XXPRJ_XXWV_FK_I'
CREATE INDEX XXPRJ_XXWV_FK_I ON XXPRJ
 (XXWV_XXWV_ID)
/
PROMPT Creating Index 'XXKP_XXKPADR_FK_I'
CREATE INDEX XXKP_XXKPADR_FK_I ON XXKP
 (XXKPADR_XXKPADR_ID)
/
PROMPT Creating Index 'XXPRJDOKU_XXPRJ_FK_I'
CREATE INDEX XXPRJDOKU_XXPRJ_FK_I ON XXPRJDOKU
 (XXPRJ_XXPRJ_ID)
/
PROMPT Creating Index 'XXVORS_XXPRJ_FK_I'
CREATE INDEX XXVORS_XXPRJ_FK_I ON XXVORS
 (XXPRJ_XXPRJ_ID)
/
PROMPT Creating Index 'XXWV_XXVORS_FK_I'
CREATE INDEX XXWV_XXVORS_FK_I ON XXWV
 (XXVORS_XXVORS_ID)
/
```

```
PROMPT Creating Index 'XXWV_XXFOL_FK_I'
CREATE INDEX XXWV_XXFOL_FK_I ON XXWV
 (XXFOL_XXFOL_ID)
/

PROMPT Creating Index 'XXWV_XXPRJ_FK_I'
CREATE INDEX XXWV_XXPRJ_FK_I ON XXWV
 (XXPRJ_XXPRJ_ID)
/

PROMPT Creating Index 'XXFOL_XXPRJ_FK_I'
CREATE INDEX XXFOL_XXPRJ_FK_I ON XXFOL
 (XXPRJ_XXPRJ_ID)
/
```

Schlüsselattribute

```
 PROMPT Creating Primary Key on 'XXPRJ'
ALTER TABLE XXPRJ
 ADD CONSTRAINT XXPRJ_PK PRIMARY KEY
  (XXPRJ_ID)
/
PROMPT Creating Primary Key on 'XXKP'
ALTER TABLE XXKP
 ADD CONSTRAINT XXKP_PK PRIMARY KEY
  (XXKP_ID)
/
PROMPT Creating Primary Key on 'XXPRJDOKU'
ALTER TABLE XXPRJDOKU
 ADD CONSTRAINT XXPRJDOKU_PK PRIMARY KEY
  (XXPRJDOKU_ID)
/
PROMPT Creating Primary Key on 'XXKONT'
ALTER TABLE XXKONT
 ADD CONSTRAINT XXKONT_PK PRIMARY KEY
  (XXKONT_ID)
/

PROMPT Creating Primary Key on 'XXVORS'
ALTER TABLE XXVORS
 ADD CONSTRAINT XXVORS_PK PRIMARY KEY
  (XXVORS_ID)
/
PROMPT Creating Primary Key on 'XXFB'
ALTER TABLE XXFB
 ADD CONSTRAINT XXFB_PK PRIMARY KEY
  (XXFB_ID)
/
PROMPT Creating Primary Key on 'XXKPADR'
ALTER TABLE XXKPADR
 ADD CONSTRAINT XXKPADR_PK PRIMARY KEY
  (XXKPADR_ID)
/
PROMPT Creating Primary Key on 'XXWV'
ALTER TABLE XXWV
 ADD CONSTRAINT XXWV_PK PRIMARY KEY
  (XXWV_ID)
/
PROMPT Creating Primary Key on 'XXFOL'
ALTER TABLE XXFOL
```

```
 ADD CONSTRAINT XXFOL_PK PRIMARY KEY
  (XXFOL_ID)
/
PROMPT Creating Unique Keys on 'XXPRJ'
ALTER TABLE XXPRJ
 ADD ( CONSTRAINT XXPRJ_PRJ_NUMMER_UK UNIQUE
  (PROJEKT_NUMMER))
/
PROMPT Creating Unique Keys on 'XXVORS'
ALTER TABLE XXVORS
 ADD ( CONSTRAINT XXVORS_XXVORS_UK UNIQUE
  (DATUM))
/
PROMPT Creating Unique Keys on 'XXWV'
ALTER TABLE XXWV
 ADD ( CONSTRAINT XXWV_XXWV_UK UNIQUE
  (DATUM))
/

PROMPT Creating Unique Keys on 'XXFOL'
ALTER TABLE XXFOL
 ADD ( CONSTRAINT XXFOL_XXFOL_UK UNIQUE
  (DATUM))
/

PROMPT Creating Check Constraints on 'XXKP'
ALTER TABLE XXKP
 ADD CONSTRAINT AVCON_83260_GESCH_000 CHECK (GESCHLECHT IN ('Herr',  'Frau',
'unbekannt'))
/

PROMPT Creating Check Constraints on 'XXKONT'
ALTER TABLE XXKONT
 ADD CONSTRAINT AVCON_83257_KONTA_000 CHECK (KONTAKTGRUPPE
IN ('eigene Initiative',  'caesar',  'Wissenschaftlicher Beirat',
'Stifter',  'Wissenschaft',  'Industrie'))
 ADD CONSTRAINT AVCON_83257_KONTA_001 CHECK (KONTAKTART
IN ('Telefonat',  'Schriftwechsel',  'Besuch',  'Dienstreise',
'caesarium'))
/

PROMPT Creating Check Constraints on 'XXVORS'
ALTER TABLE XXVORS
 ADD CONSTRAINT AVCON_83275_VORSE_000 CHECK (VORSELEKTIONSBESCHLUSS
IN ('Projekt wird weiter verfolgt',  'Projekt wird zurückgestellt',
'Projekt wird nicht weiterverfolgt'))
/

PROMPT Creating Check Constraints on 'XXFB'
ALTER TABLE XXFB
 ADD CONSTRAINT AVCON_83251_BEREI_000 CHECK (BEREICH
IN ('Materialwissenschaften',  'Koppelung elektr und biol System',
'Kommunikationsergonomie'))
 ADD CONSTRAINT AVCON_83251_PROJE_000 CHECK (PROJEKTART IN ('Pilotprojekt',
'Sonderprojekt',  'Sonstiges'))
/
PROMPT Creating Foreign Keys on 'XXPRJ'
ALTER TABLE XXPRJ ADD CONSTRAINT
 XXPRJ_XXWV_FK FOREIGN KEY
  (XXWV_XXWV_ID) REFERENCES XXWV
```

```
  (XXWV_ID) ADD CONSTRAINT
 XXPRJ_XXKONT_FK FOREIGN KEY
  (XXKONT_XXKONT_ID) REFERENCES XXKONT
  (XXKONT_ID) ADD CONSTRAINT
 XXPRJ_XXFB_FK FOREIGN KEY
  (XXFB_XXFB_ID) REFERENCES XXFB
  (XXFB_ID) ADD CONSTRAINT
 XXPRJ_XXKP_FK FOREIGN KEY
  (XXKP_XXKP_ID) REFERENCES XXKP
  (XXKP_ID)
/

PROMPT Creating Foreign Keys on 'XXKP'
ALTER TABLE XXKP ADD CONSTRAINT
 XXKP_XXKPADR_FK FOREIGN KEY
  (XXKPADR_XXKPADR_ID) REFERENCES XXKPADR
  (XXKPADR_ID)
/
PROMPT Creating Foreign Keys on 'XXPRJDOKU'
ALTER TABLE XXPRJDOKU ADD CONSTRAINT
 XXPRJDOKU_XXPRJ_FK FOREIGN KEY
  (XXPRJ_XXPRJ_ID) REFERENCES XXPRJ
  (XXPRJ_ID)
/
PROMPT Creating Foreign Keys on 'XXVORS'
ALTER TABLE XXVORS ADD CONSTRAINT
 XXVORS_XXPRJ_FK FOREIGN KEY
  (XXPRJ_XXPRJ_ID) REFERENCES XXPRJ
  (XXPRJ_ID)
/

PROMPT Creating Foreign Keys on 'XXWV'
ALTER TABLE XXWV ADD CONSTRAINT
 XXWV_XXVORS_FK FOREIGN KEY
  (XXVORS_XXVORS_ID) REFERENCES XXVORS
  (XXVORS_ID) ADD CONSTRAINT
 XXWV_XXFOL_FK FOREIGN KEY
  (XXFOL_XXFOL_ID) REFERENCES XXFOL
  (XXFOL_ID) ADD CONSTRAINT
 XXWV_XXPRJ_FK FOREIGN KEY
  (XXPRJ_XXPRJ_ID) REFERENCES XXPRJ
  (XXPRJ_ID)
/

PROMPT Creating Foreign Keys on 'XXFOL'
ALTER TABLE XXFOL ADD CONSTRAINT
 XXFOL_XXPRJ_FK FOREIGN KEY
  (XXPRJ_XXPRJ_ID) REFERENCES XXPRJ
  (XXPRJ_ID)
/
```

Sequencen

```
PROMPT Creating Sequence 'XXWV_SEQ'
CREATE SEQUENCE XXWV_SEQ
 NOMAXVALUE
 NOMINVALUE
 NOCYCLE
 NOCACHE
/
```

```
PROMPT Creating Sequence 'XXFOL_SEQ'
CREATE SEQUENCE XXFOL_SEQ
 NOMAXVALUE
 NOMINVALUE
 NOCYCLE
 NOCACHE
/
PROMPT Creating Sequence 'XXKONT_SEQ'
CREATE SEQUENCE XXKONT_SEQ
 NOMAXVALUE
 NOMINVALUE
 NOCYCLE
 NOCACHE
/

PROMPT Creating Sequence 'XXPRJ_SEQ'
CREATE SEQUENCE XXPRJ_SEQ
 NOMAXVALUE
 NOMINVALUE
 NOCYCLE
 NOCACHE
/

PROMPT Creating Sequence 'XXKP_SEQ'
CREATE SEQUENCE XXKP_SEQ
 NOMAXVALUE
 NOMINVALUE
 NOCYCLE
 NOCACHE
/

PROMPT Creating Sequence 'XXPRJDOKU_SEQ'
CREATE SEQUENCE XXPRJDOKU_SEQ
 NOMAXVALUE
 NOMINVALUE
 NOCYCLE
 NOCACHE
/

PROMPT Creating Sequence 'XXFB_SEQ'
CREATE SEQUENCE XXFB_SEQ
 NOMAXVALUE
 NOMINVALUE
 NOCYCLE
 NOCACHE
/

PROMPT Creating Sequence 'XXVORS_SEQ'
CREATE SEQUENCE XXVORS_SEQ
 NOMAXVALUE
 NOMINVALUE
 NOCYCLE
 NOCACHE
/


DELETE FROM CG_REF_CODES
WHERE RV_DOMAIN = 'ANREDE'
/
```

INSERT INTO CG_REF_CODES (RV_DOMAIN, RV_LOW_VALUE, RV_HIGH_VALUE, RV_ABBREVIATION, RV_MEANING) VALUES ('ANREDE', 'Herr', NULL, NULL, NULL)
/
INSERT INTO CG_REF_CODES (RV_DOMAIN, RV_LOW_VALUE, RV_HIGH_VALUE, RV_ABBREVIATION, RV_MEANING) VALUES ('ANREDE', 'Frau', NULL, NULL, NULL)
/

COMMIT
/

Tabellen für die Bewertung der Projektideen

```
CREATE TABLE XXBEW_DURCH (
  XXBEW_DURCH_ID      NUMBER (10)   NOT NULL,
  FK_XXPRJ_ID         NUMBER (10),
  DURCH_1             VARCHAR2 (10),
  DURCH_2             VARCHAR2 (10),
  DURCH_3             VARCHAR2 (10),
  DURCH_4             VARCHAR2 (10),
  DURCH_5             VARCHAR2 (10),
  DURCH_6             VARCHAR2 (10),
  DURCH_7             VARCHAR2 (10),
  SUMME_DURCH_NEIN    NUMBER (10),
  SUMME_DURCH_JA      NUMBER (10),
  CREATED_BY          NUMBER (15),
  CREATION_DATE       DATE          DEFAULT SYSDATE NOT NULL,
  LAST_UPDATED_BY     NUMBER (15),
  LAST_UPDATED_DATE   DATE          DEFAULT SYSDATE NOT NULL,
  LAST_UPDATED_LOGIN  NUMBER (15),
  CONSTRAINT XXBEW_DURCH_PK
  PRIMARY KEY ( XXBEW_DURCH_ID ) ) ;

CREATE TABLE XXBEW_EIG_WISS (
  XXBEW_EIG_WISS_ID   NUMBER (10)   NOT NULL,
  FK_XXPRJ_ID         NUMBER (10),
  WISS_EIG_1          NUMBER (10),
  WISS_EIG_GEW_1      NUMBER (10),
  WISS_SUM_EIG_1      NUMBER (10),
  WISS_EIG_2          NUMBER (10),
  WISS_EIG_GEW_2      NUMBER (10),
  WISS_SUM_EIG_2      NUMBER (10),
  SUMME_EIG           NUMBER (10),
  SUMME_EIG_SUM       NUMBER (10),
  CREATED_BY          NUMBER (15),
  CREATION_DATE       DATE          DEFAULT SYSDATE NOT NULL,
  LAST_UPDATED_BY     NUMBER (15),
  LAST_UPDATED_DATE   DATE          DEFAULT SYSDATE NOT NULL,
  LAST_UPDATED_LOGIN  NUMBER (15),
  CONSTRAINT XXBEW_EIG_WISS_PK
  PRIMARY KEY ( XXBEW_EIG_WISS_ID ) ) ;

CREATE TABLE XXBEW_RISIKO (
  XXBEW_RISIKO_ID     NUMBER (10)   NOT NULL,
  FK_XXPRJ_ID         NUMBER (10),
  RISIKO_GEW_1        NUMBER (10),
  RISIKO_1            NUMBER (10),
  RISIKO_SUM_1        NUMBER (10),
  RISIKO_PRO_1        NUMBER (10,3),
  RISIKO_GEW_2        NUMBER (10),
  RISIKO_2            NUMBER (10),
  RISIKO_SUM_2        NUMBER (10),
  RISIKO_PRO_2        NUMBER (10,3),
  RISIKO_GEW_3        NUMBER (10),
  RISIKO_3            NUMBER (10),
  RISIKO_SUM_3        NUMBER (10),
```

```
  RISIKO_PRO_3        NUMBER (10,3),
  RISIKO_GEW_4        NUMBER (10),
  RISIKO_4            NUMBER (10),
  RISIKO_SUM_4        NUMBER (10),
  RISIKO_PRO_4        NUMBER (10,3),
  RISIKO_GEW_5        NUMBER (10),
  RISIKO_5            NUMBER (10),
  RISIKO_SUM_5        NUMBER (10),
  RISIKO_PRO_5        NUMBER (10,3),
  SUMME_RISIKO        NUMBER (10),
  SUMME_RISIKO_SUM    NUMBER (10),
  CREATED_BY          NUMBER (15),
  CREATION_DATE       DATE          DEFAULT SYSDATE NOT NULL,
  LAST_UPDATED_BY     NUMBER (15),
  LAST_UPDATED_DATE   DATE          DEFAULT SYSDATE NOT NULL,
  LAST_UPDATED_LOGIN  NUMBER (15),
  CONSTRAINT XXBEW_RISIKO
  PRIMARY KEY ( XXBEW_RISIKO_ID ) ) ;

CREATE TABLE XXBEW_WIRT (
  XXBEW_WIRT_ID       NUMBER (10)   NOT NULL,
  FK_XXPRJ_ID         NUMBER (10),
  WIRT_1              NUMBER (10),
  WIRT_2              NUMBER (10),
  WIRT_3              NUMBER (10),
  WIRT_4              NUMBER (10),
  WIRT_5              NUMBER (10),
  WIRT_6              NUMBER (10),
  WIRT_7              NUMBER (10),
  WIRT_8              NUMBER (10),
  WIRT_9              NUMBER (10),
  WIRT_10             NUMBER (10),
  SUMME_WIRT          NUMBER (10,3),
  WIRT_GEW_1          NUMBER (10),
  WIRT_GEW_2          NUMBER (10),
  WIRT_GEW_3          NUMBER (10),
  WIRT_GEW_4          NUMBER (10),
  WIRT_GEW_5          NUMBER (10),
  WIRT_GEW_6          NUMBER (10),
  WIRT_GEW_7          NUMBER (10),
  WIRT_GEW_8          NUMBER (10),
  WIRT_GEW_9          NUMBER (10),
  WIRT_SUM_1          NUMBER (10),
  WIRT_SUM_10         NUMBER (10),
  WIRT_SUM_2          NUMBER (10),
  WIRT_SUM_3          NUMBER (10),
  WIRT_SUM_4          NUMBER (10),
  WIRT_SUM_5          NUMBER (10),
  WIRT_SUM_6          NUMBER (10),
  WIRT_SUM_7          NUMBER (10),
  WIRT_SUM_8          NUMBER (10),
  WIRT_SUM_9          NUMBER (10),
  WIRT_GEW_10         NUMBER (10),
  SUMME_WIRT_SUM      NUMBER (10,3),
  CREATED_BY          NUMBER (15),
  CREATION_DATE       DATE          DEFAULT SYSDATE NOT NULL,
  LAST_UPDATED_BY     NUMBER (15),
  LAST_UPDATED_DATE   DATE          DEFAULT SYSDATE NOT NULL,
  LAST_UPDATED_LOGIN  NUMBER (15),
  CONSTRAINT XXBEW_WIRT_PK
  PRIMARY KEY ( XXBEW_WIRT_ID ) ) ;

CREATE TABLE XXBEW_WISS (
  XXBEW_WISS_ID       NUMBER (10)   NOT NULL,
  FK_XXPRJ_ID         NUMBER (10),
  WISS_1              NUMBER (10),
  WISS_2              NUMBER (10),
  WISS_3              NUMBER (10),
  WISS_4              NUMBER (10),
```

```
  WISS_5              NUMBER (10),
  WISS_6              NUMBER (10),
  WISS_7              NUMBER (10),
  WISS_8              NUMBER (10),
  WISS_9              NUMBER (10),
  WISS_10             NUMBER (10),
  SUMME_WISS          NUMBER (10),
  WISS_GEW_1          NUMBER (10),
  WISS_GEW_2          NUMBER (10),
  WISS_GEW_3          NUMBER (10),
  WISS_GEW_4          NUMBER (10),
  WISS_GEW_5          NUMBER (10),
  WISS_GEW_6          NUMBER (10),
  WISS_GEW_7          NUMBER (10),
  WISS_GEW_8          NUMBER (10),
  WISS_GEW_9          NUMBER (10),
  WISS_SUM_1          NUMBER (10),
  WISS_SUM_10         NUMBER (10),
  WISS_SUM_2          NUMBER (10),
  WISS_SUM_3          NUMBER (10),
  WISS_SUM_4          NUMBER (10),
  WISS_SUM_5          NUMBER (10),
  WISS_SUM_6          NUMBER (10),
  WISS_SUM_7          NUMBER (10),
  WISS_SUM_8          NUMBER (10),
  WISS_SUM_9          NUMBER (10),
  WISS_GEW_10         NUMBER (10),
  SUMME_WISS_SUM      NUMBER (10),
  CREATED_BY          NUMBER (15),
  CREATION_DATE       DATE          DEFAULT SYSDATE NOT NULL,
  LAST_UPDATED_BY     NUMBER (15),
  LAST_UPDATED_DATE   DATE          DEFAULT SYSDATE NOT NULL,
  LAST_UPDATED_LOGIN  NUMBER (15),
  CONSTRAINT XXBEW_WISS_PK
  PRIMARY KEY ( XXBEW_WISS_ID ) ) ;
```

Sequencen für die Bewertungs-Tabellen

```
CREATE SEQUENCE XXBEW_DURCH_SEQ
NOMAXVALUE
 NOMINVALUE
 NOCYCLE
 NOCACHE
/
CREATE SEQUENCE XXBEW_EIG_WISS_SEQ
NOMAXVALUE
 NOMINVALUE
 NOCYCLE
 NOCACHE
/
CREATE SEQUENCE XXBEW_RISIKO_SEQ
NOMAXVALUE
 NOMINVALUE
 NOCYCLE
 NOCACHE
/
CREATE SEQUENCE XXBEW_WIRT_SEQ
NOMAXVALUE
 NOMINVALUE
 NOCYCLE
 NOCACHE
/
CREATE SEQUENCE XXBEW_WISS_SEQ
```

```
NOMAXVALUE
 NOMINVALUE
 NOCYCLE
 NOCACHE
/
```

CG_REF_CODES

Tabelle für die Domänen

```
DELETE FROM CG_REF_CODES
WHERE RV_DOMAIN = 'FB_BEREICH'
/
INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)       VALUES ('FB_BEREICH',  '1',  NULL,  NULL,
'Materialwissenschaften')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)       VALUES ('FB_BEREICH',  '2',  NULL,  NULL,  'Koppelung
elektronischer und biologischer Systeme')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)       VALUES ('FB_BEREICH',  '3',  NULL,  NULL,
'Kommunikationsergonomie')
/
COMMIT
/


DELETE FROM CG_REF_CODES
WHERE RV_DOMAIN = 'PRJ_ART'
/

INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)       VALUES ('PRJ_ART',  'Pilotprojekt',  NULL,  NULL,
'Pilotprojekt')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)       VALUES ('PRJ_ART',  'Sonstiges',  NULL,  NULL,  'Sonstiges')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)       VALUES ('PRJ_ART',  'Sonderprojekt',  NULL,  NULL,
'Sonderprojekt')
/
COMMIT
/
DELETE FROM CG_REF_CODES
WHERE RV_DOMAIN = 'VORSELEKTIONSBESCHLUSS'
/
INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)       VALUES ('VORSELEKTIONSBESCHLUSS',  'weiter verfolgt',
NULL,  NULL,  'weiter verfolgt')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)       VALUES ('VORSELEKTIONSBESCHLUSS',  'zurückgestellt',
NULL,  NULL,  'zurückgestellt')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)       VALUES ('VORSELEKTIONSBESCHLUSS',  'wird nicht
weiterverfolgt',  NULL,  NULL,  'wird nicht weiterverfolgt')
```

```
/
COMMIT
/

DELETE FROM CG_REF_CODES
WHERE RV_DOMAIN = 'GEWFAKT_PRJPOT'
/

INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)       VALUES ('GEWFAKT_PRJPOT',  '10',  NULL,  NULL,  '10')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)       VALUES ('GEWFAKT_PRJPOT',  '0',  NULL,  NULL,  '0')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)       VALUES ('GEWFAKT_PRJPOT',  '1',  NULL,  NULL,  '1')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)       VALUES ('GEWFAKT_PRJPOT',  '2',  NULL,  NULL,  '2')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)       VALUES ('GEWFAKT_PRJPOT',  '3',  NULL,  NULL,  '3')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)       VALUES ('GEWFAKT_PRJPOT',  '4',  NULL,  NULL,  '4')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)
VALUES ('GEWFAKT_PRJPOT',  '5',  NULL,  NULL,  '5')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)
VALUES ('GEWFAKT_PRJPOT',  '6',  NULL,  NULL,  '6')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)
VALUES ('GEWFAKT_PRJPOT',  '7',  NULL,  NULL,  '7')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)
VALUES ('GEWFAKT_PRJPOT',  '8',  NULL,  NULL,  '8')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)
VALUES ('GEWFAKT_PRJPOT',  '9',  NULL,  NULL,  '9')
/

COMMIT
/

DELETE FROM CG_REF_CODES
WHERE RV_DOMAIN = 'KONTAKTART'
/

INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)
VALUES ('KONTAKTART',  'Besuch',  NULL,  NULL,  'Besuch')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN,  RV_LOW_VALUE,  RV_HIGH_VALUE,
RV_ABBREVIATION,  RV_MEANING)
```

```
VALUES ('KONTAKTART', 'Dienstreise', NULL, NULL, 'Dienstreise')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN, RV_LOW_VALUE, RV_HIGH_VALUE,
RV_ABBREVIATION, RV_MEANING)
VALUES ('KONTAKTART', 'caesarium', NULL, NULL, 'caesarium')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN, RV_LOW_VALUE, RV_HIGH_VALUE,
RV_ABBREVIATION, RV_MEANING)
VALUES ('KONTAKTART', 'Telefonat', NULL, 'Telefonat', 'Telefonat')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN, RV_LOW_VALUE, RV_HIGH_VALUE,
RV_ABBREVIATION, RV_MEANING)
VALUES ('KONTAKTART', 'Vorbesprechung', NULL, 'Vorbesprechung', 'Vorbesprechung')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN, RV_LOW_VALUE, RV_HIGH_VALUE,
RV_ABBREVIATION, RV_MEANING)
VALUES ('KONTAKTART', 'Schriftwechsel', NULL, NULL, 'Schriftwechsel')
/

COMMIT
/

DELETE FROM CG_REF_CODES
WHERE RV_DOMAIN = 'KGRART'
/

INSERT INTO CG_REF_CODES (RV_DOMAIN, RV_LOW_VALUE, RV_HIGH_VALUE,
RV_ABBREVIATION, RV_MEANING)
VALUES ('KGRART', 'eigene Initiative', NULL, NULL, 'eigene Initiative')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN, RV_LOW_VALUE, RV_HIGH_VALUE,
RV_ABBREVIATION, RV_MEANING)
VALUES ('KGRART', 'CAESAR', NULL, 'CAESAR', 'CAESAR')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN, RV_LOW_VALUE, RV_HIGH_VALUE,
RV_ABBREVIATION, RV_MEANING)
VALUES ('KGRART', 'Wissenschaftlicher Beirat', NULL, 'Wissenschaftlicher B', 'Wissenschaftlicher
Beirat')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN, RV_LOW_VALUE, RV_HIGH_VALUE,
RV_ABBREVIATION, RV_MEANING)
VALUES ('KGRART', 'Stifter', NULL, 'Stifter', 'Stifter')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN, RV_LOW_VALUE, RV_HIGH_VALUE,
RV_ABBREVIATION, RV_MEANING)
VALUES ('KGRART', 'Wissenschaft', NULL, 'Wissenschaft', 'Wissenschaft')
/
INSERT INTO CG_REF_CODES (RV_DOMAIN, RV_LOW_VALUE, RV_HIGH_VALUE,
RV_ABBREVIATION, RV_MEANING)
VALUES ('KGRART', 'Industrie', NULL, 'Industrie', 'Industrie')
/

COMMIT
/
```

Paket AD_DD

AD_DD.register_table (registrieren der Tabellen)

```
execute ad_dd.register_table('INNOC', 'XXPRJ', 'T', 8, 10, 90);
execute ad_dd.register_table('INNOC', 'XXKONT', 'T', 8, 10, 90);
```

```
execute ad_dd.register_table('INNOC', 'XXWV', 'T', 8, 10, 90);
execute ad_dd.register_table('INNOC', 'XXKP', 'T', 8, 10, 90);
execute ad_dd.register_table('INNOC', 'XXVORS', 'T', 8, 10, 90);
execute ad_dd.register_table('INNOC', 'XXFOL', 'T', 8, 10, 90);
execute ad_dd.register_table('INNOC', 'XXFB', 'T', 8, 10, 90);
execute ad_dd.register_table('INNOC', 'XXPRJDOKU', 'T', 8, 10, 90);
execute ad_dd.register_table('INNOC', 'XXKONT', 'T', 8, 10, 90);
```

ADD_DD Register_Column (registrieren der Spalten)

XXPRJ

execute ad_dd.register_column('INNOC', 'XXPRJ', 'XXPRJ_ID', '1', NUMBER', '10', 'N', 'N');
execute ad_dd.register_column('INNOC', 'XXPRJ', 'TITEL', '2', 'VARCHAR2', '100', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXPRJ', 'ERST_DATUM', '4', 'DATE', '', 'N', 'N');
execute ad_dd.register_column('INNOC', 'XXPRJ', 'BESCHREIBUNG', '5', 'VARCHAR2', '1000', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXPRJ', 'CREATED_BY', '6', 'NUMBER', '15', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXPRJ', 'CREATION_DATE', '7', 'DATE', '9', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXPRJ', 'LAST_UPDATED_BY', '8', 'NUMBER', '15', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXPRJ', 'LAST_UPDATED_DATE', '9', 'DATE', '9', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXPRJ', 'LAST_UPDATED_LOGIN', '10', 'NUMBER', '15', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXPRJ', 'XXFB_ID', '13', 'NUMBER', '13', 'Y', 'N');

XXKONT

execute ad_dd.register_column('INNOC', 'XXKONT', 'XXKONT_ID', '1', 'NUMBER', '10', 'N', 'N');
execute ad_dd.register_column('INNOC', 'XXKONT', ' KONTAKTGRUPPE, '2', 'VARCHAR2', '240', 'Y', 'N')
execute ad_dd.register_column('INNOC', 'XXKONT', ' KONTAKTART, '3', 'VARCHAR2', '240', 'Y', 'N')
execute ad_dd.register_column('INNOC', 'XXKONT', 'CREATED_BY', '4', 'NUMBER', '15', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXKONT', 'CREATION_DATE', '5', 'DATE', '9', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXKONT', 'LAST_UPDATED_BY', '6', 'NUMBER', '15', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXKONT', 'LAST_UPDATED_DATE', '7', 'DATE', '9', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXKONT', 'LAST_UPDATED_LOGIN', '8', 'NUMBER', '15', 'Y', 'N');

XXWV

execute ad_dd.register_column('INNOC', 'XXWV', 'XXERSTDOK_ID', '1', 'NUMBER10', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXWV', 'DATUM', '2', 'DATE', '', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXWV', 'VERTEILER', '3', 'VARCHAR2', '100', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXWV', 'ANLAGEN', '4', 'VARCHAR2', '100', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXWV', 'BEARBEITER', '5', 'VARCHAR2', '100', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXWV', 'VERSAND_PRJANTRAG', '6', 'VARCHAR2', '20', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXWV', 'CREATED_BY', '7', 'NUMBER', '15', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXWV', 'CREATION_DATE', '8', 'DATE', '9', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXWV', 'LAST_UPDATED_BY', '9', 'NUMBER', '15', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXWV', 'LAST_UPDATED_DATE', '10', 'DATE', '9', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXWV', 'LAST_UPDATED_LOGIN', '11', 'NUMBER', '15', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXWV', 'XXFOL_ID', '12', 'NUMBER', '10', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXWV', 'XXVORS_ID', '13', 'NUMBER', '10', 'N', 'N');

XXKP

execute ad_dd.register_column('INNOC', 'XXKP', 'XXKP_ID', '1', 'NUMBER', '10', 'N', 'N');
execute ad_dd.register_column('INNOC', 'XXKP', 'ANREDE', '2', 'VARCHAR2', '240', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXKP', 'NAME', '3', 'VARCHAR2', '100', 'N', '
execute ad_dd.register_column('INNOC', 'XXKP', 'STRASSE', '4', 'VARCHAR2', '100', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXKP', 'PLZ', '5', 'NUMBER', '10', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXKP', 'ORT', '6', 'VARCHAR2', '100', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXKP', 'TELEFON', '7', 'VARCHAR2', '100', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXKP', 'EMAIL', '8', 'VARCHAR2', '100', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXKP', 'CREATED_BY', '9', 'NUMBER', '15', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXKP', 'CREATION_DATE', '10', 'DATE', '9', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXKP', 'LAST_UPDATED_BY', '11', 'NUMBER', '15', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXKP', 'LAST_UPDATED_DATE', '12', 'DATE', '9', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXKP', 'LAST_UPDATED_LOGIN', '13', 'NUMBER', '15', 'Y', 'N');

XXERSTDOK

execute ad_dd.register_column('INNOC', 'XXERSTDOK', 'XXERSTDOK_ID', '1', 'NUMBER', '10', 'N', 'N');
execute ad_dd.register_column('INNOC', ' GESPRAECHSNOTIZ', 'VARCHAR2, '2', 'DATE', '400', 'N', 'N');
execute ad_dd.register_column('INNOC', 'XXERSTDOK', 'XXKGR_ID', '3', 'NUMBER', '10', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXERSTDOK', 'XXKA_ID', '4', 'NUMBER', '10', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXERSTDOK', 'XXKP_ID', '5', 'NUMBER', '10', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXERSTDOK', 'XXPRJ_ID', '6', 'NUMBER', '10', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXERSTDOK', 'CREATED_BY', '7', 'NUMBER', '15', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXERSTDOK', 'CREATION_DATE', '8', 'DATE', '9', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXERSTDOK', 'LAST_UPDATED_BY', '9', 'NUMBER', '15', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXERSTDOK', 'LAST_UPDATED_DATE', '10', 'DATE', '9', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXERSTDOK', 'LAST_UPDATED_LOGIN', '11', 'NUMBER', '15', 'Y', 'N');

XXVORS

execute ad_dd.register_column('INNOC', 'XXVORS', 'XXVORS_ID', '1', 'NUMBER', '10', 'N', 'N');
execute ad_dd.register_column('INNOC', 'XXVORS', 'DATUM', '2', 'DATE', '', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXVORS', 'BEGRÜNDUNG', '3', 'VARCHAR2', '400', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXVORS', 'PRJ_WEITER', '4', 'VARCHAR2', '100', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXVORS', 'CREATED_BY', '5', 'NUMBER', '15', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXVORS', 'CREATION_DATE', '6', 'DATE', '9', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXVORS', 'LAST_UPDATED_BY', '7', 'NUMBER', '15', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXVORS', 'LAST_UPDATED_DATE', '8', 'DATE', '9', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXVORS', 'LAST_UPDATED_LOGIN', '9', 'NUMBER', '15', 'Y', 'N');

XXFOL

execute ad_dd.register_column('INNOC', 'XXFOL', 'XXFOL_ID', '1', 'NUMBER', '10', 'N', 'N');
execute ad_dd.register_column('INNOC', 'XXFOL', 'DATUM', '2', 'DATE', '9', 'N', 'N');
execute ad_dd.register_column('INNOC', 'XXFOL', 'GESPRÄCHSNOTIZ', '3', 'VARCHAR2', '400', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXFOL', 'VEREINBARTE_AKTIVITÄTEN', '4', 'VARCHAR2', '400', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXFOL', 'XXPRJ_ID', '5', 'NUMBER', '10', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXFOL', 'CREATED_BY', '6', 'NUMBER', '15', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXFOL', 'CREATION_DATE', '7', 'DATE', '9', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXFOL', 'LAST_UPDATED_BY', '8', 'NUMBER', '15', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXFOL', 'LAST_UPDATED_DATE', '9', 'DATE', '9', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXFOL', 'LAST_UPDATED_LOGIN', '10', 'NUMBER', '15', 'Y', 'N');

XXFB

execute ad_dd.register_column('INNOC', 'XXFB', 'XXFB_ID', '1', 'NUMBER', '10', 'N', 'N');
execute ad_dd.register_column('INNOC', 'XXFB', 'DISZIPLIN', '2', 'VARCHAR2', '240', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXFB', 'THEMENKREIS', '3', 'VARCHAR2', '240', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXFB', 'BEREICH', '4', 'VARCHAR2', '240', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXFB', 'FB_ART', '5', 'VARCHAR2', '240', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXFB', 'TRIPLET', '6', 'VARCHAR2', '240', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXFB', 'CREATED_BY', '7', 'NUMBER', '15', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXFB', 'CREATION_DATE', '8', 'DATE', '9', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXFB', 'LAST_UPDATED_BY', '9', 'NUMBER', '15', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXFB', 'LAST_UPDATED_DATE', '10', 'DATE', '9', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXFB', 'LAST_UPDATED_LOGIN', '11', 'NUMBER', '15', 'Y', 'N');

XXPRJDOKU

execute ad_dd.register_column('INNOC', 'XXPRJDOKU', 'XXPRJDOKU_ID', '1', 'NUMBER', '10', 'N', 'N');

```
execute ad_dd.register_column('INNOC', 'XXPRJDOKU', 'PRJ_ANTRAG_DATUM', '2', 'DATE', '9', 'Y',
'N');
execute ad_dd.register_column('INNOC', 'XXPRJDOKU', 'PRJ_ANTRAG_RUECK_DATUM', '3', 'DATE',
'9', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXPRJDOKU', 'PRJ_ANTRAG_DATEIANHANG', '4', 'DATE',
'9', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXPRJDOKU', 'BESCHLUS_DATUM', '5', 'DATE', '9', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXPRJDOKU', 'BESCHLUS_KURZTEXT', '6', 'VARCHAR2',
'200', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXPRJDOKU', 'GUTACHTEN_DATUM', '7', 'DATE', '9', 'Y',
'N');
execute ad_dd.register_column('INNOC', 'XXPRJDOKU', 'GUTACHTEN_BESCHREIBUNG', '8',
'VARCHAR2', '200', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXPRJDOKU', 'XXPRJ_ID', '9', 'NUMBER', '10', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXPRJDOKU', 'CREATED_BY', '10', 'NUMBER', '15', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXPRJDOKU', 'CREATION_DATE', '11', 'DATE', '9', 'Y', 'N');
execute ad_dd.register_column('INNOC', 'XXPRJDOKU', 'LAST_UPDATED_BY', '12', 'NUMBER', '15', 'Y',
'N');
execute ad_dd.register_column('INNOC', 'XXPRJDOKU', 'LAST_UPDATED_DATE', '13', 'DATE', '9', 'Y',
'N');
execute ad_dd.register_column('INNOC', 'XXPRJDOKU', 'LAST_UPDATED_LOGIN', '14', 'NUMBER',
'15', 'Y', 'N');
```

Grants an den User APPS vergeben

```
GRANT ALTER ON        XXPRJ   TO APPS WITH GRANT OPTION;
GRANT DELETE ON       XXPRJ   TO APPS WITH GRANT OPTION;
GRANT INDEX ON        XXPRJ   TO APPS WITH GRANT OPTION;
GRANT INSERT ON       XXPRJ   TO APPS WITH GRANT OPTION;
GRANT SELECT ON       XXPRJ   TO APPS WITH GRANT OPTION;
GRANT UPDATE ON       XXPRJ   TO APPS WITH GRANT OPTION;
GRANT REFERENCES ON   XXPRJ   TO APPS WITH GRANT OPTION;


GRANT ALTER ON        XXKGR     TO APPS WITH GRANT OPTION;
GRANT DELETE ON       XXKGR     TO APPS WITH GRANT OPTION;
GRANT INDEX ON        XXKGR     TO APPS WITH GRANT OPTION;
GRANT INSERT ON       XXKGR     TO APPS WITH GRANT OPTION;
GRANT SELECT ON       XXKGR     TO APPS WITH GRANT OPTION;
GRANT UPDATE ON       XXKGR     TO APPS WITH GRANT OPTION;
GRANT REFERENCES ON   XXKGR     TO APPS WITH GRANT OPTION;

GRANT ALTER ON        XXWV      TO APPS WITH GRANT OPTION;
GRANT DELETE ON       XXWV      TO APPS WITH GRANT OPTION;
GRANT INDEX ON        XXWV      TO APPS WITH GRANT OPTION;
GRANT INSERT ON       XXWV      TO APPS WITH GRANT OPTION;
GRANT SELECT ON       XXWV      TO APPS WITH GRANT OPTION;
GRANT UPDATE ON       XXWV      TO APPS WITH GRANT OPTION;
GRANT REFERENCES ON   XXWV      TO APPS WITH GRANT OPTION;

GRANT ALTER ON        XXWV   TO APPS WITH GRANT OPTION;
GRANT DELETE ON       XXWV   TO APPS WITH GRANT OPTION;
GRANT INDEX ON        XXWV   TO APPS WITH GRANT OPTION;
GRANT INSERT ON       XXWV   TO APPS WITH GRANT OPTION;
GRANT SELECT ON       XXWV   TO APPS WITH GRANT OPTION;
GRANT UPDATE ON       XXWV   TO APPS WITH GRANT OPTION;
GRANT REFERENCES ON   XXWV   TO APPS WITH GRANT OPTION;

GRANT ALTER ON        XXKP      TO APPS WITH GRANT OPTION;
```

```
GRANT DELETE ON      XXKP          TO APPS WITH GRANT OPTION;
GRANT INDEX ON       XXKP          TO APPS WITH GRANT OPTION;
GRANT INSERT ON      XXKP          TO APPS WITH GRANT OPTION;
GRANT SELECT ON      XXKP          TO APPS WITH GRANT OPTION;
GRANT UPDATE ON      XXKP          TO APPS WITH GRANT OPTION;
GRANT REFERENCES ON        XXKP          TO APPS WITH GRANT OPTION;

GRANT ALTER ON       XXKA          TO APPS WITH GRANT OPTION;
GRANT DELETE ON      XXKA    TO APPS WITH GRANT OPTION;
GRANT INDEX ON       XXKA    TO APPS WITH GRANT OPTION;
GRANT INSERT ON      XXKA    TO APPS WITH GRANT OPTION;
GRANT SELECT ON      XXKA    TO APPS WITH GRANT OPTION;
GRANT UPDATE ON      XXKA    TO APPS WITH GRANT OPTION;
GRANT REFERENCES ON        XXKA    TO APPS WITH GRANT OPTION;

GRANT ALTER ON       XXERSTDOK     TO APPS WITH GRANT OPTION;
GRANT DELETE ON      XXERSTDOK     TO APPS WITH GRANT OPTION;
GRANT INDEX ON       XXERSTDOK     TO APPS WITH GRANT OPTION;
GRANT INSERT ON      XXERSTDOK     TO APPS WITH GRANT OPTION;
GRANT SELECT ON      XXERSTDOK     TO APPS WITH GRANT OPTION;
GRANT UPDATE ON      XXERSTDOK     TO APPS WITH GRANT OPTION;
GRANT REFERENCES ON        XXERSTDOK     TO APPS WITH GRANT OPTION;

GRANT ALTER ON       XXVORS        TO APPS WITH GRANT OPTION;
GRANT DELETE ON      XXVORS        TO APPS WITH GRANT OPTION;
GRANT INDEX ON       XXVORS        TO APPS WITH GRANT OPTION;
GRANT INSERT ON      XXVORS        TO APPS WITH GRANT OPTION;
GRANT SELECT ON      XXVORS        TO APPS WITH GRANT OPTION;
GRANT UPDATE ON      XXVORS        TO APPS WITH GRANT OPTION;
GRANT REFERENCES ON        XXVORS        TO APPS WITH GRANT OPTION;

GRANT ALTER ON       XXFOL   TO APPS WITH GRANT OPTION;
GRANT DELETE ON      XXFOL  TO APPS WITH GRANT OPTION;
GRANT INDEX ON       XXFOL  TO APPS WITH GRANT OPTION;
GRANT INSERT ON      XXFOL  TO APPS WITH GRANT OPTION;
GRANT SELECT ON      XXFOL  TO APPS WITH GRANT OPTION;
GRANT UPDATE ON      XXFOL  TO APPS WITH GRANT OPTION;
GRANT REFERENCES ON        XXFOL  TO APPS WITH GRANT OPTION;

GRANT ALTER ON       XXFB          TO APPS WITH GRANT OPTION;
GRANT DELETE ON      XXFB          TO APPS WITH GRANT OPTION;
GRANT INDEX ON       XXFB          TO APPS WITH GRANT OPTION;
GRANT INSERT ON      XXFB          TO APPS WITH GRANT OPTION;
GRANT SELECT ON      XXFB          TO APPS WITH GRANT OPTION;
GRANT UPDATE ON      XXFB          TO APPS WITH GRANT OPTION;
GRANT REFERENCES ON        XXFB          TO APPS WITH GRANT OPTION;

GRANT ALTER ON       XXPRJDOKU     TO APPS WITH GRANT OPTION;
GRANT DELETE ON      XXPRJDOKU     TO APPS WITH GRANT OPTION;
GRANT INDEX ON       XXPRJDOKU     TO APPS WITH GRANT OPTION;
GRANT INSERT ON      XXPRJDOKU     TO APPS WITH GRANT OPTION;
GRANT SELECT ON      XXPRJDOKU     TO APPS WITH GRANT OPTION;
GRANT UPDATE ON      XXPRJDOKU     TO APPS WITH GRANT OPTION;
GRANT REFERENCES ON        XXPRJDOKU     TO APPS WITH GRANT OPTION;
```

Grants für die Sequencen

```
GRANT SELECT ON XXFOL_SEQ TO APPS WITH GRANT OPTION;
GRANT ALTER ON XXFOL_SEQ TO APPS WITH GRANT OPTION;
GRANT SELECT ON XXKONT_SEQ TO APPS WITH GRANT OPTION;
GRANT ALTER ON XXKONT_SEQ TO APPS WITH GRANT OPTION;
GRANT SELECT ON XXPRJ_SEQ TO APPS WITH GRANT OPTION;
GRANT ALTER ON XXPRJ_SEQ TO APPS WITH GRANT OPTION;
GRANT SELECT ON XXKP_SEQ TO APPS WITH GRANT OPTION;
GRANT ALTER ON XXKP_SEQ TO APPS WITH GRANT OPTION;
GRANT SELECT ON XXPRJDOKU_SEQ TO APPS WITH GRANT OPTION;
GRANT ALTER ON XXPRJDOKU_SEQ TO APPS WITH GRANT OPTION;
GRANT SELECT ON XXERSTDOK_SEQ TO APPS WITH GRANT OPTION;
GRANT ALTER ON XXERSTDOK_SEQ TO APPS WITH GRANT OPTION;
GRANT SELECT ON XXFB_SEQ TO APPS WITH GRANT OPTION;
GRANT ALTER ON XXFB_SEQ TO APPS WITH GRANT OPTION;
GRANT SELECT ON XXVORS_SEQ TO APPS WITH GRANT OPTION;
GRANT ALTER ON XXVORS_SEQ TO APPS WITH GRANT OPTION;
```

Reports:

1	Erstdokumentation	innoc_XXPRJ
2	Folgedokumentation	innoc_Folgedokumente
3	Vorselektion	innoc_Vorselektion
4	Projektvorlauf	Innocon_XXPROJEKTVORLAUF
5	Projektverlaufzeitanalyse	Projektverlaufzeitanalyse

1. Erstdokumentation

Der Report "Erstdokumentation“ besteht aus 5 Abfragen und einer Parametermaske:

```
Parametermaske:
 XX_Projektnummer:
 select xxprj_id, titel
 from innocon.xxprj
  WHERE XXPRJ.FLAG IS NULL
ORDER BY XXPRJ_ID DESC

1. G_XXProjekt:
SELECT XXPRJ.XXPRJ_ID,
 XXPRJ.TITEL,
 XXPRJ.DATUM  PRJ_DATUM,
 XXPRJ.BESCHREIBUNG,
 XXPRJ.GESPRAECHSNOTIZ_ERSTDOK,
 XXPRJ.GEW_GRUNDLAGENFORSCHUNG,
 XXPRJ.GEW_ANWENDUNG,
 XXPRJ.GEW_ARBEITSMARKT,
 XXPRJ.GEW_PRESTIGE
FROM INNOCON.XXPRJ
WHERE XXPRJ_ID =: XX_Projektnummer;

2. Kont
SELECT
  XXKONT.KONTAKTGRUPPE,
  XXKONT.SONSTIGE_KGR,
  XXKONT.KONTAKTART,
  XXKONT.SONSTIGE_KA,
  XXKONT.XXKONT_ID
FROM INNOCON.XXKONT, INNOCON.XXPRJ
WHERE XXPRJ_ID =: XX_Projektnummer AND
 XXPRJ.FK_XXKONT_ID = XXKONT.XXKONT_ID

3. Kontaktperson:
SELECT XXKP.NAME
FROM INNOCON.XXKP, INNOCON.XXPRJ
WHERE XXPRJ_ID =: XX_Projektnummer AND
 XXKP.FK_XXPRJ_ID = XXPRJ.XXPRJ_ID


4. Wiedervorlage:
SELECT
 XXWV.DATUM WVDATUM,
 XXWV.VERTEILER_AN,
 XXWV.ANLAGEN,
 XXWV.BEARBEITER,
```

```
 XXWV.VERSAND_PRJANTRAG,
 XXWV.FK_XXPRJ_ID,
 XXWV.XXWV_ID
FROM INNOCON.XXWV
```

5. XXFB:

```
SELECT XXFB.TRIPLET,
 XXFB.DISZIPLIN,
 XXFB.THEMENKREIS,
 XXFB.BEREICH,
 XXFB.PROJEKTART,
 XXFB.XXFB_ID
FROM INNOCON.XXFB , INNOCON.XXPRJ
WHERE XXPRJ_ID =: XX_Projektnummer AND
XXPRJ.FK_XXFB_ID = XXFB.XXFB_ID
```

2. Folgedokumentation

1. Projektidee:

```
SELECT XXPRJ.XXPRJ_ID
, XXPRJ.TITEL
, XXPRJ.DATUM  PRJ_DATUM
, XXPRJ.BESCHREIBUNG
, XXPRJ.GESPRAECHSNOTIZ_ERSTDOK
, XXPRJ.GEW_GRUNDLAGENFORSCHUNG
, XXPRJ.GEW_ANWENDUNG
,XXPRJ.GEW_ARBEITSMARKT
, XXPRJ.GEW_PRESTIGE
FROM
 INNOCON.XXPRJ,  INNOCON.XXFOL
WHERE
 XXPRJ.FLAG is null
AND  XXPRJ.XXPRJ_ID =  XXFOL.FK_XXPRJ_ID
ORDER BY XXPRJ_ID DESC
```

2. Kontaktperson:

```
SELECT XXKP.NAME,
 XXKP.FK_XXPRJ_ID FK_XXPRJ_ID
FROM INNOCON.XXKP, INNOCON.XXPRJ
WHERE  XXKP.FK_XXPRJ_ID = XXPRJ.XXPRJ_ID
```

3. Folgedokumente:

```
SELECT XXFOL.DATUM   Folgedok_Datum,
XXFOL.VEREINBARTE_AKTIVITAETEN,
XXFOL.GESPRAECHSNOTIZ XXFOL_NOTIZ,
XXFOL.FK_XXPRJ_ID,
XXFOL.XXFOL_ID
FROM INNOCON.XXFOL, INNOCON.XXPRJ
WHERE XXFOL.FK_XXPRJ_ID = XXPRJ.XXPRJ_ID
ORDER BY XXFOL.DATUM
```

4. Wiedervorlage:

```
SELECT
 XXWV.DATUM WVDATUM,
 XXWV.VERTEILER_AN,
 XXWV.ANLAGEN,
 XXWV.BEARBEITER,
 XXWV.VERSAND_PRJANTRAG,
 XXWV.TASK  WV_TASK,
```

```
 XXWV.FK_FOL_ID,
 XXWV.FK_XXPRJ_ID,
XXWV.XXWV_ID
FROM INNOCON.XXWV , innocon.XXPRJ, innocon.XXFOL
WHERE
 XXWV.FK_FOL_ID = XXFOL.XXFOL_ID
 AND  XXWV.FK_XXPRJ_ID = XXPRJ.XXPRJ_ID
```

3. Vorselektion

```
SELECT   XXPRJ.TITEL
, XXPRJ.DATUM
, XXPRJ.PRJ_NUMMER
, XXKP.NAME
, XXKONT. KONTAKTGRUPPE
, XXKONT.KONTAKTART
,XXPRJ.BESCHREIBUNG
, XXPRJPOT.GEW_GRUNDLAGENFORSCHUNG
, XXPRJPOT.GEW_ANWENDUNG
, XXPRJPOT.GEW_ARBEITSMARKT
, XXPRJPOT.GEW_PRESTIGE
, XXFB.THEMENKREIS
, XXFB.BEREICH
, XXFB.PROJEKT_ART
FROM INNOCON.XXPRJ,
INNOCON.XXKP, INNOCON.XXKONT,
INNOCON.XXPRJPOT,   INNOCON.XXFB
WHERE   XXPRJ.PRJ_NUMMER = XXKP.XXKP_ID
 AND       XXPRJ.PRJ_NUMMER = XXKONT.XXKONT_ID
 AND       XXPRJ.PRJ_NUMMER = XXPRJPOT.XXPRJPOT_ID
 AND       XXPRJ.PRJ_NUMMER = XXFB.XXFB_ID
```

4. Projektvorlauf

```
1.  Q_XXPRJ:
SELECT DISTINCT
  XXPRJ.XXPRJ_ID                    Projektnummer,
  XXPRJ.TITEL                       Projekttitel,
  XXPRJ.DATUM                       Erstkontakt_DATUM
 FROM  INNOCON.XXPRJ
 WHERE   INNOCON.XXPRJ.XXPRJ_ID =: XX_Projektnummer

2. XXFOL :
SELECT DISTINCT
XXFOL.DATUM Folgekontakt_DATUM,
XXFOL.FK_XXPRJ_ID
FROM INNOCON.XXFOL, INNOCON.XXPRJ
WHERE
INNOCON.XXFOL.FK_XXPRJ_ID =  XXPRJ.XXPRJ_ID
 AND INNOCON.XXPRJ.XXPRJ_ID =: XX_Projektnummer

3. Q_XXKONT:
SELECT DISTINCT
  XXKONT.KONTAKTART                 Kontaktart,
  XXKONT.SONSTIGE_KA                Sonstige_KA,
  XXKONT.KONTAKTGRUPPE     Kontagtgruppe,
  XXKONT.SONSTIGE_KGR               Sonstige_KGR,
  XXKONT.XXKONT_ID
FROM   INNOCON.XXPRJ,  INNOCON.XXKONT
WHERE INNOCON.XXPRJ.XXPRJ_ID =: XX_Projektnummer
```

```
AND INNOCON.XXPRJ.FK_XXKONT_ID = XXKONT.XXKONT_ID

4. Q_XXPRJDOKU:
SELECT DISTINCT
XXPRJDOKU.PRJ_ANTRAG_DATUM      Projektantrag_DATUM,
XXPRJDOKU.GUTACHTEN_DATUM       Projektgutachten_DATUM,
XXPRJDOKU.BESCHLUS_DATUM        Projektbeschluss_DATUM,
XXPRJDOKU.FK_XXPRJ_ID
FROM   INNOCON.XXPRJ,  INNOCON.XXPRJDOKU
WHERE
INNOCON.XXPRJ.XXPRJ_ID =: XX_Projektnummer

5. Q_XXVORS:
SELECT DISTINCT
XXVORS.DATUM                    Vorselektion_DATUM,
XXVORS.FK_XXPRJ_ID
FROM   INNOCON.XXPRJ,  INNOCON.XXVORS
WHERE
-- AND INNOCON.XXPRJ.XXPRJ_ID =: XX_Projektnummer
INNOCON.XXVORS.FK_XXPRJ_ID = XXPRJ.XXPRJ_ID

6. Q_XXWV:
SELECT  XXWV.DATUM
, XXWV.VERTEILER_AN
, XXWV.ANLAGEN
, XXWV.BEARBEITER,
XXWV.VERSAND_PRJANTRAG
, XXWV.FK_XXPRJ_ID
, XXWV.XXWV_ID
FROM INNOCON.XXWV
```

6. Projektverlaufzeitanalyse

```
1. Q_ANZ_Gutachten:
SELECT  COUNT(*)
FROM  INNOCON.XXPRJDOKU
WHERE  XXPRJDOKU.GUTACHTEN_DATUM IS NOT NULL
2. Q_ANZ_IDEE:
SELECT  count(*) FROM INNOCON.XXPRJ
3. Q_ANZ_Projektabschluss:
SELECT  count(*) FROM INNOCON.XXPRJDOKU
WHERE  XXPRJDOKU.FERTIG_DATUM is not null
4. Q_ANZ_Projektstart:
SELECT  count(*) FROM INNOCON.XXPRJDOKU
WHERE  XXPRJDOKU.BESCHLUS_DATUM is not null
5. Q_ANZ_VORS_Ablehnen:
SELECT  count(*) FROM INNOCON.XXPRJ
WHERE  XXPRJ.FLAG = 'wird nicht weiterverfolgt'
6. Q_ANZ_VORS_WEITER:
SELECT  count(*) FROM INNOCON.XXPRJ
WHERE  XXPRJ.FLAG = 'weiter verfolgt'
7. Q_Erstkontakte:
SELECT XXPRJ_ID Projektnummer,
DATUM ErstDatum, FLAG  weiterverfolgt
FROM INNOCON.XXPRJ
ORDER BY XXPRJ_ID
8. Q_G_sum_phase_I_II:
select sum(to_char(xxvors.datum - xxprj.datum)) sum_phase_I_II
from  innocon.xxprj, innocon.XXVORS
WHERE  XXVORS.FK_XXPRJ_ID = XXPRJ.XXPRJ_ID
```

```
AND XXPRJ.FLAG = 'weiter verfolgt'
9. Q_Gutachten:
SELECT  XXPRJDOKU.GUTACHTEN_DATUM Gutachten,
XXPRJDOKU.FK_XXPRJ_ID   FK_GUTACHTEN
FROM INNOCON.XXPRJ, INNOCON.XXPRJDOKU
WHERE XXPRJDOKU.FK_XXPRJ_ID = XXPRJ.XXPRJ_ID
AND XXPRJDOKU.GUTACHTEN_DATUM is not null
10. Q_PHASE_I_II:
select xxprj.datum estdatum,
xxvors.datum vorsdatum,
xxvors.fk_xxprj_ID  FKPhase_I_II,
to_char(xxvors.datum - xxprj.datum) PHASE_I_II
from innocon.xxprj, innocon.xxvors
WHERE  XXPRJ.FLAG = 'weiter verfolgt'
AND   XXVORS.FK_XXPRJ_ID = XXPRJ.XXPRJ_ID
11. Q_PHASE_I_III:
select  xxprj.datum estdatum,
 xxprjdoku.GUTACHTEN_DATUM,
 to_char(xxprjdoku.GUTACHTEN_DATUM - xxprj.datum) phase_I_III,
 xxprjdoku.fk_xxprj_id FK_PHASE_I_III
from innocon.xxprj,
 innocon.XXPRJDOKU
where     XXPRJDOKU.FK_XXPRJ_ID = XXPRJ.XXPRJ_ID
AND xxprjdoku.GUTACHTEN_DATUM IS NOT NULL
12. Q_PHASE_IV:
SELECT to_char(XXPRJDOKU.BESCHLUS_DATUM  - xxprj.datum) PHASE_V,
 XXPRJDOKU.FK_XXPRJ_ID   FK_PHASE_IV
FROM INNOCON.XXPRJ, INNOCON.XXPRJDOKU
WHERE XXPRJDOKU.FK_XXPRJ_ID = XXPRJ.XXPRJ_ID
 AND XXPRJDOKU.BESCHLUS_DATUM is not null
13. Q_PHASE_V:
select to_char(XXPRJDOKU.FERTIG_DATUM  - xxprj.datum) PHASE_V,
 xxprjdoku.fk_xxprj_id FK_PHASE_V
FROM   INNOCON.XXPRJ,  INNOCON.XXPRJDOKU
where XXPRJDOKU.FK_XXPRJ_ID = XXPRJ.XXPRJ_ID AND
XXPRJDOKU.FERTIG_DATUM is not null
14. Q_Projektabschluss:
SELECT XXPRJDOKU.FERTIG_DATUM  Projektabschluss,
XXPRJDOKU.FK_XXPRJ_ID   FK_PRJAbschluss
FROM INNOCON.XXPRJ, INNOCON.XXPRJDOKU
WHERE XXPRJDOKU.FK_XXPRJ_ID = XXPRJ.XXPRJ_ID AND
XXPRJDOKU.FERTIG_DATUM is not null
15. Q_Projektstart:
SELECT XXPRJDOKU.BESCHLUS_DATUM Projektstart,
 XXPRJDOKU.FK_XXPRJ_ID   FK_PRJSTART
FROM INNOCON.XXPRJ, INNOCON.XXPRJDOKU
WHERE XXPRJDOKU.FK_XXPRJ_ID = XXPRJ.XXPRJ_ID
AND XXPRJDOKU.BESCHLUS_DATUM is not null
16. Q_SUM_PHASE_I_III:
select sum(to_char(xxprjdoku.GUTACHTEN_DATUM - xxprj.datum)) sum_phase_I_III
from  innocon.xxprj, innocon.XXPRJDOKU
where  XXPRJDOKU.FK_XXPRJ_ID = XXPRJ.XXPRJ_ID
AND xxprjdoku.GUTACHTEN_DATUM IS NOT NULL
17. Q_SUM_PHASE_IV:
SELECT sum(to_char(XXPRJDOKU.BESCHLUS_DATUM  - xxprj.datum) ) SUM_PHASE_V
FROM INNOCON.XXPRJ, INNOCON.XXPRJDOKU
WHERE XXPRJDOKU.FK_XXPRJ_ID = XXPRJ.XXPRJ_ID
AND XXPRJDOKU.BESCHLUS_DATUM is not null
18. Q_SUM_PHASE_V:
select sum(to_char(XXPRJDOKU.FERTIG_DATUM  - xxprj.datum) ) SUM_PHASE_V
```

FROM INNOCON.XXPRJ, INNOCON.XXPRJDOKU
where XXPRJDOKU.FK_XXPRJ_ID = XXPRJ.XXPRJ_ID AND
XXPRJDOKU.FERTIG_DATUM is not null
19. Q_VORS_Ablehenen:
SELECT XXVORS.DATUM VORS_DATUM_ABLEHNEN, XXVORS.FK_XXPRJ_ID FK_ABL_ID
FROM INNOCON.XXPRJ, INNOCON.XXVORS
WHERE XXPRJ.FLAG = 'wird nicht weiterverfolgt'
AND XXVORS.FK_XXPRJ_ID = XXPRJ.XXPRJ_ID
20. Q_VORS_WEITER:
SELECT XXVORS.DATUM VORS_DATUM_WEITER,
XXVORS.FK_XXPRJ_ID FK_Weiter_ID
FROM INNOCON.XXPRJ,
INNOCON.XXVORS
WHERE XXPRJ.FLAG = 'weiter verfolgt' AND
XXVORS.FK_XXPRJ_ID = XXPRJ.XXPRJ_ID

Realisierung der Anhänge

Um die Funktion ‚Verknüpfungen' mit einer Maske zu verbinden, werden Daten in folgende Tabellen des Datenbankbenutzers APPLSYS eingefügt.

1. FND_ATTACHMENT_FUNCTIONS
2. FND_DOC_CATEGORY_USAGES
3. FND_ATTACHMENT_BLOCKS
4. FND_ATTACHMENT_BLK_ENTITIES

Wichtig dabei ist die Bezeichnung der Forms-Maske in *Oracle Applications*. Hier wird nur die Syntax für die Maske XX_PRJ (Projektideen) dargestellt.

FND_FORM_FUNCTIONS		FND_FORM	
FUNCTION_ID	FUNCTION_NAME	FORM_NAME	FORM_ID
INNOC_XX_PRJ	4073	XX_PRJ	55247
INNOC_XXVORS	4093	XX_VORS	55268
INNOC_XXFOL	4113	XX_FOL	55308
INNOC_XXPRJBEW	4133	XX_PRJBEW	55328

1. FND_ATTACHMENT_FUNCTIONS:

```
INSERT INTO FND_ATTACHMENT_FUNCTIONS (
  ATTACHMENT_FUNCTION_ID
, FUNCTION_TYPE
, FUNCTION_ID
, FUNCTION_NAME
, CREATION_DATE
, CREATED_BY
, LAST_UPDATE_DATE
, LAST_UPDATED_BY
, LAST_UPDATE_LOGIN
, APPLICATION_ID
, SESSION_CONTEXT_FIELD
,  ENABLED_FLAG    )
SELECT
  FND_ATTACHMENT_FUNCTIONS_S.nextval
, func.FUNCTION_TYPE
, form.form_id
, form.form_name
,  sysdate
, 1
, sysdate
, 1
, null
, form.application_id
, func.SESSION_CONTEXT_FIELD
, func.ENABLED_FLAG
FROM  fnd_form form, fnd_attachment_functions func
WHERE       form.form_name = 'XX_PRJ'
          and   func.function_name  = 'INNOC_XX_PRJ'
```

2. FND_DOC_CATEGORY_USAGES

```
INSERT INTO FND_DOC_CATEGORY_USAGES (
DOC_CATEGORY_USAGE_ID
, CATEGORY_ID
, ATTACHMENT_FUNCTION_ID
, ENABLED_FLAG
, CREATION_DATE
, CREATED_BY
, LAST_UPDATE_DATE
, LAST_UPDATED_BY
, LAST_UPDATE_LOGIN )
SELECT
FND_DOC_CATEGORY_USAGES_S.nextval
, doc.CATEGORY_ID
, func1.attachment_function_id
, doc.enabled_flag
, sysdate
, 1
, sysdate
, 1
, null
FROM
 FND_DOC_CATEGORY_USAGES doc
, fnd_attachment_functions func1,
WHERE  func1.function_name = 'INNOC_XX_PRJ'
 AND   func1.attachment_function_id = doc.attachment_function_id
```

3. FND_ATTACHMENT_BLOCKS

```
INSERT INTO FND_ATTACHMENT_BLOCKS
( ATTACHMENT_BLK_ID
, ATTACHMENT_FUNCTION_ID
, BLOCK_NAME
, QUERY_FLAG
, SECURITY_TYPE
, CREATION_DATE
, CREATED_BY
, LAST_UPDATE_DATE
, LAST_UPDATED_BY
, LAST_UPDATE_LOGIN
, CONTEXT1_FIELD )
values (
FND_ATTACHMENT_BLOCKS_S.nextval
, 1000019
, 'XXPRJ'
,'Y'
, 4
, sysdate
, 1
, sysdate
, 1
, 1
, 'XX_PRJ.XXPRJ_ID' )
```

4. FND_ATTACHMENT_BLK_ENTITIES

```
INSERT INTO FND_ATTACHMENT_BLK_ENTITIES(
  ATTACHMENT_BLK_ENTITY_ID
, ATTACHMENT_BLK_ID
, DISPLAY_METHOD
, INCLUDE_IN_INDICATOR_FLAG
, CREATION_DATE
, CREATED_BY
, LAST_UPDATE_DATE
, LAST_UPDATED_BY
, LAST_UPDATE_LOGIN
, PK1_FIELD
, PK2_FIELD
, PK3_FIELD
, PK4_FIELD
, PK5_FIELD
, SQL_STATEMENT
, DATA_OBJECT_CODE
, INDICATOR_IN_VIEW_FLAG
, QUERY_PERMISSION_TYPE
, INSERT_PERMISSION_TYPE
, UPDATE_PERMISSION_TYPE
, DELETE_PERMISSION_TYPE
, CONDITION_FIELD
, CONDITION_OPERATOR
, CONDITION_VALUE1
, CONDITION_VALUE2 )
SELECT
  FND_ATTACHMENT_BLK_ENTITIES_S.nextval
, b2.ATTACHMENT_BLK_ID
, 'R'
, 'Y'
, sysdate
, 1
, sysdate
, 1 , 1 , 'XXPRJ.XXPRJ_ID'
, NULL
, NULL
, NULL
, NULL
, NULL
, 'XXPRJ'
, 'N'
, 'Y'
, 'T'
, 'T'
, 'T'
, 'XXPRJ.XXPRJ_ID'
, 1
, 'Y'
, NULL
FROM
  fnd_ATTACHMENT_BLOCKS b2
, fnd_attachment_functionsf2
WHERE   b2.attachment_function_id = f2.attachment_function_id
 and f2.function_name = 'INNOC_XX_PRJ'
 and b2.block_name = 'XX_PRJ'
```